CONCURRENT CONTROLS, INC.

We Built Our Name On Multiuser Solutions™

CCI Concurrent DOS V3.06

Users Guide

CCI Concurrent DOS V3.06 Users Guide

Notice

Concurrent Controls, Inc. shall not be liable for technical or editorial errors, or omissions contained herein, nor shall Concurrent Controls, Inc. be liable for incidental or consequential damages resulting from the furnishing, performance or use of this material. See the accompanying Software License Agreement for full details and disclosure of warranties and limitations as may be contained therein.

This manual should not be construed as any representation or warranty with respect to the software named herein. Concurrent Controls, Inc. reserves the right to make changes in its technical information and specifications. Generally, if such changes or variations are known to exist and to affect the Software Product significantly, a release note or README file accompanies the manual and the associated Software disks.

The Software described in this user manual is furnished under a license agreement or nondisclosure agreement. The Software may be used or copied only in accordance with the terms of the accompanying Software License Agreement.

Concurrent Controls, Inc., makes no representations or implied warranties of merchantability or fitness for any particular purpose herein as related to the Software or hardware referenced within this document or as may be necessary to implement the Software product described herein.

Copyright © 1994 Concurrent Controls, Inc.
All rights reserved.

Except as permitted by such license, this document is provided with the understanding that it shall not be copied or reproduced in whole or part, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, recording or otherwise disclosed to others without the prior written permission of Concurrent Controls.

The Concurrent Controls logo, We Built Our Name On Multiuser Solutions™, CCI Concurrent DOS™, CCI TERM™, CCI NET™ and eMAIL™ are trademarks of Concurrent Controls, Inc. Carbon Copy™ is a trademark of Meridian Technology and Microcom, Inc. Novell® and NetWare® are registered trademarks of Novell Inc. Pentium® is a registered trdemark of Intel Corporation. PC-Technician™ is a trademark of Windsor Technologies. Touchtone® is a registered trademark of AT&T. VGA MaxStation™ is a trademark of Maxpeed Corporation. WordStar® is a registered trademark of MicroPro International Corporation. WordPerfect® is a registered trademark of Wordperfect Corporation.

Printed in the United States of America.

Table of Contents

Chapter1: Introduction 1–1

Glossary 1–1
Operating System Concepts 1–2
Multitasking 1–3
Multiuser 1–4
Memory 1–4
Storage Media 1–4
Files 1–5
File Attributes 1–6
File and Path Passwords 1–7
Wildcards 1–7
Drives 1–8
Floating Drives 1–8
Current Drive 1–8
Pathing 1–9
Changing Removable Media 1–9
CCI Concurrent DOS Commands 1–10
Command Line Operation 1–14
Function Keys 1–16
Redirection 1–16
Piping 1–17
On-Line Help 1–17

Chapter 2: Installation and Configuration 2–1

Installation Checklist and Hardware Review 2–1
Pre-Installation Considerations 2–3
Installation 2–4
Operating System Configuration 2–5
Configuring the Host Computer and System Console 2–6
Configuring COM1 and COM2 2–6
Configuring System Memory 2–8
Setting System Console Sessions and System Reboot Capabilities 2–8
Customizing System Features in SETUP 2–9
Configuring Multiuser Serial Ports and Display Adapters 2–9
Multiuser Display Adapters — VGA MaxStation 2–10
Device Setup 2–10
Handshaking 2–11
Terminal Setup 2–11
PC/AT Terminal Setup 2–12
Modem Setup 2–12
Printer Setup 2–13
Mouse Setup 2–13

Troubleshooting Mouse Problems 2–14
Finished Setup 2–14
Customizing the System 2–14
Changing or Adding Hardware 2–17
Optimizing for Best Performance 2–17

Chapter 3: Batch File Processing 3–1

Types of Batch Files 3–1
CCIBOOT Batch File 3–1
STARTxxy Batch Files 3–1
NAME Batch Files 3–2
LOGIN Batch Files 3–2
Other Batch Files 3–2
Creating a Batch File 3–3
Running Batch Files 3–3
Environment 3–3
&SETCCI and &CLEARCCI 3–4
Batch File Variables 3–8
Batch Commands 3–11

Chapter 4: Security System 4–1

Enabling Security 4–1
User Parameters and Password Maintenance 4–2
CONTROL Configuration File 4–6
INCLUDE and EXCLUDE 4–7
Boot-time options 4–8
Login Options 4–8
Reporting Options 4–9
System Administration Log Options 4–10
Using the File Security System 4–11
Using the Security System 4–17

Chapter 5: Remote Access, Support and Communication 5–1

Remote Access 5–1
Modem, Remote Workstations and Communications 5–1
Remote Access Quick Set-up 5–3
Setting Up the Ports On the Host Computer 5–4
Using the Same Modem for Dial-in and Dial-out 5–4
Modem Configuration 5–4
Programming the Remote and Dial-in Modems 5–5
Multitech V32 9600 Baud Modem Initialization 5–5
Cardinal 9600 BPD V.42BIS Modem Initialization 5–7
Practical Peripherals Modem Initialization 5–9

Zoom VFX V32bis Modem Initialization 5–10
U.S. Robotics Courier Modem Initialization 5–11
U.S. Robotics Sportster Modem Initialization 5–13
INTEL EX Modem Initialization 5–14
Most Commonly Used AT Commands For Modems 5–16
Troubleshooting Modem Problems 5–17
Monitoring Remote Channel Status 5–19
Remote Support 5–25
The User/Device that is Being Taken Over 5–25
How That User or Device is Being Taken Over 5–26
Ending the TAKEOVER session 5–27
TRANSIT 5–28
TRANSIT Basics 5–28
TRANSIT Script Language 5–32
TRANSIT Script Examples 5–40
TRANSIT Script ERROR Codes 5–44

Chapter 6: Scheduler System 6–1

Scheduling Jobs (SCHEDULE) 6–1
Executing Jobs (JOBMGR) 6–4

Chapter 7: Menuing System 7–1

Files, Structures, and Requirements 7–1
Terminal Data Files 7–2
Built-in Commands 7–2
When external commands are used 7–3
Building Menus — EDITMENU 7–3
EDITMENU Options 7–4
Compressing a Menu — COPYMENU 7–6
Error Messages 7–7

Chapter 8: Print Spooler System 8–1

CSPOOL Commands 8–2
Configuring Applications for the Spooler 8–8
CSPOOL Installation 8–9
Automatically Starting the Spooler 8–10
Changing the Spool File directory 8–10
The Spooler Configuration File 8–11
CSPOOL Error Messages 8–15

Index ..... page a

Chapter 1: Introduction

Introduction

An operating system is software that controls the computer's hardware, software, peripherals and input/output system. It also controls how programs are executed by the computer. CCI Concurrent DOS is operating system software for computers using the Intel 80386, 80486, Pentium, and compatible processors.

CCI Concurrent DOS™ can be used in conjunction with CCI NET™, a peer-to-peer networking product that connects two or more CCI Concurrent DOS systems.

Because this system is a multiuser DOS solution, it provides the advantages of using the familiar DOS interface and command set. In addition, CCI Concurrent DOS has built-in multitasking, which allows the execution of multiple applications at the same time. Many other capabilities are also included in the CCI Concurrent DOS operating system, which will be discussed in this manual.

The minimum hardware required to use CCI Concurrent DOS is a microcomputer using the appropriate processor, 4MB of RAM, and a 40 MB hard disk.

This first chapter in this manual defines the terminology used when discussing the multiuser, multitasking environment. The second chapter covers step-by-step installation and configuration. The third chapter covers the creation and use of batch files. System security is covered in the fourth chapter. The fifth chapter reviews the built-in remote access subsystem. The Scheduler subsystem is covered in the sixth chapter. The menuing subsystem is covered in the seventh chapter, and the last chapter summarizes the print spooler subsystem.

Glossary

This manual uses a number of terms that deal with multiuser, multitasking concepts:

Multiuser display adapter (MDA): a special interface board used to connect multiple VGA monitors and standard keyboards to a computer. Special CCI Concurrent DOS drivers are available for these boards.

MDA stations: monitors and keyboards attached to MDAs.

Multiuser serial boards: an interface board supporting 2 to 64 RS-232 or RS-422 serial ports used to connect text-based workstations to a computer. These ports are sometimes called add-on serial ports because they are usually in addition to the standard PC COM1 and COM2 serial ports. A non-intelligent (or standard) serial board just contains serial ports and is

managed by the computer's main processor. An intelligent serial port board has its own on-board processor and memory, relieving the main processor of much of the work involved in managing the ports.

PCTERM Stations: the PC terminals and or PC's running PCTERM emulation software and connected either locally or remotely via modem, to COM ports and/or multiuser serial boards.

Host computer: the computer running CCI Concurrent DOS and acting as the server, or "host," for multiple text and graphics workstations.

Host console: the monitor and keyboard directly connected to the host computer.

Workstation: a text or graphics based display and keyboard combination connected to the host computer via a Multiuser Display Adapter or a Multiuser Serial Port Board.

Session: a specific DOS activity. The host console and most workstations can support multiple DOS sessions. Pressing a special key combination enables the user to hotkey among the sessions running on their workstation. The session that is currently displayed is called the foreground or current session. All sessions not being displayed (but possibly still executing) are called background sessions.

Workstation printer: some workstations have an extra serial or parallel port. CCI Concurrent DOS supports a workstation printer connected to either of those ports. Workstation printers can be used by any user or session on the system.

Operating System Concepts

Any operating system has major functions that include:

- Controlling attached peripherals, such as the computer's video monitor, hard disk, and printer.
- Managing files that are used and/or created by the user and/or the user's applications.
- Running applications programs.

Furthermore, the CCI Concurrent DOS system offers many additional useful and unique features, including:

- Multiuser support, allowing several users (from 1 to 67) to share the power and resources of a single computer.
- Multitasking, allowing several programs to execute at once. Each user station can run as many as four programs at the same time.
- High speed serial connection support, with speeds up to 115 Kbps.

- Disk caching, to eliminate hard disk bottlenecks.
- Extended memory support, so that up to 64MB of extended RAM can be used by the system; 128MB on EISA systems.
- Security shell, to restrict access and control user environments.
- Customizable command shell that allows the deletion or modification of any built-in or external command on a per user basis.
- Extended environment variable set, so that batch files can determine if they are running in the foreground or the background, their display and driver type, user name and number, printer number, etc.
- Electronic mail, to pass messages and text files from one user to another, eliminating phone tag and paperwork.
- Enhanced remote access, to provide secure access and complete functionality for remote workstations.
- Remote support so that installation, maintenance diagnostic, and training functions can be performed from a remote location, eliminating onsite travel (similar to Carbon Copy™ functions).
- Support for the system administrator to be able to share and/or monitor any display, either local or remote, on the system.
- PCTERM terminal emulator, with autodial, screen refresh, file transfer capabilities, and full color text support.
- Built-in file transfer so that files can be transferred from a remote station to the host computer, and a script language to automate these functions.
- Job scheduling system so that programs or tasks can be automatically executed by the operating system on the date and time specified.

Multitasking

CCI Concurrent DOS is a pre-emptive multitasking operating system. This means that multiple programs, or processes, are in memory at the same time and appear to be running simultaneously. Internally, the operating system is using a pre-emptive time-slicing technique to accomplish this. Each program and process receives a slice of the processor time, running for up to 1/60 of a second. Then control is turned over to the next program or process, which gets its time slice and so on. Multiple sessions means that multiple tasks occur on each station. Other sessions may be displayed by depressing the CTRL key and the 1, 2, 3 or 4 key on the keypad.

Multiuser

The CCI Concurrent DOS operating system allows the connection of additional workstations, using either text-based terminals connected to serial ports or graphics monitors and keyboards connected through a graphics display adapter. All of the workstations support up to four multitasking sessions as well. When additional workstations are connected, the computer's power and resources can be shared among those stations, with each appearing to be an independent computer. The CCI Concurrent DOS system runs most DOS programs by using the virtual 8086 mode and other features of the 80386 (or 80486 or Pentium) processor to make the non-intelligent workstations appear to be independent computers. To DOS applications, a CCI Concurrent DOS system appears to be a network, so that the application executes transparently.

Memory

The CCI Concurrent DOS software is responsible for managing computer memory. Memory can be broken into several categories: conventional, high, extended, and expanded (EMS). CCI Concurrent DOS manages the conventional memory, which runs from 0 to 640K in the system. Whenever possible, the high memory region (which runs from 640K to the 1 MB line) is used to bank portions of the operating system into, creating a larger space for applications to run.

CCI Concurrent DOS uses the extended RAM in the computer both for its own use and to emulate EMS or LIM memory. It does this by using the processor's paged memory manager and other features. Extended RAM is generally referred to as paged memory because portions of this memory are paged in and out as needed by the programs and processes running.

Storage Media

DOS media includes floppy disks, hard disks, virtual disks, and other storage systems, such as WORM drives or CD ROMs. The media can be formatted under DOS or under any of the CCI Concurrent formatting utilities. Files stored on this media support the use of passwords and time stamping. The hard disk partitions made by CCI Concurrent DOS are equivalent to DOS V4.X or V5.X partitions. DOS media created by DOS versions 2.X through 6.2 are supported as well.

The utilities used to format, query, and use DOS storage media with CCI Concurrent DOS system are identical to, and in many cases, enhanced versions of the analogous utilities used in MS- DOS or PC-DOS. Therefore, users of any version of DOS are already familiar with many of the commands used in CCI Concurrent DOS.

The DOS media uses a directory structure that resembles a tree, with its root beneath the ground, and the trunk of the tree spreading out into many major and minor branches as it reaches higher. The DOS media has a root directory, and from that, one or more levels of subdirectories.

There are several commands that are related to the directory structure, including MKDIR, RMDIR, CHDIR, DIR, XDIR and TREE. MKDIR (or MD) creates a subdirectory, and RMDIR (or RD) removes a subdirectory. CHDIR (or CD) changes the current directory, and displays the current directory path. DIR and XDIR display the files in the current or specified directory. DIR displays the files and their size; XDIR shows the files, their size, attributes, and time and date stamp. TREE displays a listing of all files by directory and reflects the hierarchical directory structure, listing each subdirectory and all subdirectories that are located within it.

Files

File naming guidelines in the CCI Concurrent DOS system are identical to those in DOS. Filenames can be one to eight characters long, with an optional extension that can be one to three characters long. Filenames can use both letters and numbers. The characters below should not be used in filenames:

```
= , . ; : / ~ ! @ $ & * ( ) [ ] { } \ /
```

Filename extensions vary by program type; some of the more commonly seen file extensions include the following:

BAK	backup file
BAS	BASIC programming language file
BAT	batch file
CMD	Older CCI Concurrent DOS command file
COM	DOS or CCI Concurrent DOS command file
DAT	data file
EXE	DOS program file
HLP	help file
SYS	driver or operating system file
TXT	ASCII text file

File Attributes

Each file has a set of file attributes used by the operating system to determine the ways that a file can be accessed. File attributes are changed using the ATTRIB command or XATTRIB (if file security is active). Files can have any of the following attributes:

- Archive (A)
- Read-only (R)
- System (S)
- Hidden (H)
- Write (W) (only with file security)
- Execute (E) (only with file security)
- Delete (D) (only with file security)

The archive attribute is used to determine whether or not a file has been modified. When a change is made to a file, the archive attribute is automatically set to ON. Backup programs, like the BACKUP program or most tape backup software, have options to allow only copying files with archive attributes set to ON. Once the files are backed up, the archive attribute is reset to OFF. This means that incremental backups can be made with only those files that have been changed since the last backup was performed.

A file with the read-only attribute ON cannot be modified or deleted, although it may be renamed or copied. A file with the read-only attribute set to OFF is called a read-write file, and is the default mode when a file is created.

The system attribute determines whether or not files are seen in the directory. If the system attribute is ON, some programs (like DIR and COPY) can not see the file in the directory and so can not display or access them. When files are first created, the system attribute is usually set to OFF. Files with the system attribute OFF are called directory files because the DIR command can display them. Most programs have command line switches that enable them to see system files (like DIR/A or COPY/S). Other programs, like XDIR, ignore the system attribute.

The hidden attribute is much like the system attribute. Files with the hidden attribute turned ON are also not displayed by programs like DIR. Originally, the hidden attribute was supposed to hide files on the disk.

Using the file security system (described in Chapter 4) is a more effective way to protect important files against accidental deletion and control access to a file than using the system, hidden, and read-only attributes. With file security, different levels of file access can be created for different users or groups of users.

File and Path Passwords

If file security is not activated, files — either individually, groups of files, or entire subdirectories — can still be password protected. Passwords can be used to prevent the use of a file or program by unauthorized users, or to prevent the accidental deletion or modification of other files. Some of the system administration utilities — the SECURITY and TAKEOVER commands — can be password protected to limit their use.

CAUTION: *Password protected files are not protected when accessed under MS-DOS, since MS-DOS currently does not support passwords.*

The PASSWORD command is used to password protect a file. The password itself can be up to eight characters long, and can include any characters that are legal in filenames or filename extensions. The password is an optional portion of the file specification, appearing immediately after the filename and is preceded by a semicolon. In the example below, the password GENUINE is being supplied to use the TAKEOVER program.

```
TAKEOVER;GENUINE
```

If a password is not supplied, the system prompts the user to enter one. Since most application programs do not accept passwords as part of a filename, a default password can be assigned to the current session.

Wildcards

CCI Concurrent DOS supports wildcards when specifying filenames. The asterisk and question mark (* and ?) can be used in a filename or file extension; they represent a filename outline that the operating system attempts to match to other files on a disk. Wildcards can be used to indicate groups of files, such as *.TXT, meaning all files with an extension of TXT. With this, a single command line entry can be used to locate or perform a function on a group of files. The ? represents a single character and * represents one or more characters. For example, to find every file with the extension .DAT use either of the following commands:

```
DIR ????????.DAT
DIR *.DAT
```

Eight question marks are used, one for each possible character in the filename itself, while the asterisk represents anything in the zero to eight character range. To locate all of the files that begin with the letters CCI, use the following:

```
DIR CCI*.*
DIR CCI?????.*
```

To locate all of the files that begin with the letters CCI and have a file extension .TXT, use the following statement to quickly locate those files:

```
DIR CCI*.TXT
```

Drives

A drive is a storage area where files are stored. A drive can be a floppy disk, a hard disk, a memory disk, or other storage device. A floppy disk or memory disk is designated with a single letter. Since physical hard disk drives typically consist of one or more partitions, each of these partitions is designated with a unique letter.

The letters A and B are reserved for the floppy drives, and cannot be used by the hard disk. If only one floppy exists, then it is drive A: and there is no drive B:. Hard disk partitions are designated beginning with the letter C, and continuing through the letter K; the letter L is reserved for use by the operating system itself.

Floating Drives

The CCI Concurrent DOS system provides a special type of drive, called a floating drive. This can be used to make a subdirectory look like a drive. For example, if the data files for an accounting program are in the C:\GPOSS\DATA subdirectory, the floating drive I: can be defined to be the subdirectory C:\GPOSS\DATA with this command

```
CHDIR I:=C:\GPOSS\DATA
```

Then, when a DIR is done on the I: drive, the data files are displayed. The accounting program can also be configured to look at the I: drive for its data files. This is useful when a program is hard-coded for a particular drive or path.

Current Drive

Each session on a workstation has its own current, logged, or default drive. Session one may be looking at the floppy drive, while the second session is looking at (or logged into) the C: partition of the hard drive. In order to change the drive that is displayed, type the new drive letter to change to with a colon at the prompt, and depress the return key. If the C: drive is displayed, enter D: to change to the D: drive, and depress Enter or Return.

The current drive is the first place the operating system looks to find a file or program that has been requested. If it is not found in the current drive and directory, the operating system then searches each location specified in the path for the file or program requested.

Pathing

The path is a list of locations that the operating system looks sequentially to locate files or programs requested with filename extensions of .CMD, .COM, .EXE and .BAT. The APPEND command can be used to search the path for text or program overlay files.

A path statement can include multiple drives, directory and subdirectory specifications. The initial path, when the operating system first comes up, is the root directory of the boot drive. A path statement in the startup files (batch files that are automatically executed when the system powers on or reboots) can expand on that. A sample path statement is:

```
PATH C:\CDOS;C:\CDOS\MAIL;C:\WP51;D:\ACCTG
```

When accessing files, specify the file's location by preceding the filename with a drive letter, or a drive letter and subdirectory specification. By doing so, that drive and specific location is the only one searched for that file, and other locations specified in the path are not checked. If no drive is specified, then the current drive and path is checked in the order described above.

When accessing program and batch files, the filename extension may be specified in the command line. If omitted, then the operating system searches for the filename and extensions in the following order:

CMD	Native mode program file
COM	DOS command file
EXE	DOS executable file
BAT	batch file

If the filename is not found with a CMD extension, then the search continues for the filename with a COM extension, and so on, down the list. When the list is exhausted, then the operating system moves to the next location in the path and runs through the list again, searching for the filename specified, and checking for the filename extensions in the order specified above. For example, SAMPLE.EXE will be found and executed before SAMPLE.BAT is found, if they are both in the same location. To execute SAMPLE.BAT, specify the filename extension (that is, SAMPLE.BAT instead of just SAMPLE) in the command line.

Changing Removable Media

Removable media refers to either floppy disks or removable hard disk media. In either case, confirm that the disk light is not on before removing it. In addition, ensure that no files on the removable media are open when it is removed.

CAUTION: *The integrity of the data may be damaged if the files are not closed or the disk light is ON before removing the removable media.*

CCI Concurrent DOS Commands

This operating system is both multiuser and DOS-based in look, feel and command set. Many of the CCI Concurrent DOS commands for installation, maintenance and execution of programs are identical or enhanced versions of DOS commands. Because this system provides far more capabilities than DOS, there are a number of programs and commands that are new. The table on the next page lists the DOS command set and the corresponding command in CCI Concurrent DOS. Commands that differ are in italics.

DOS Commands vs. CCI Concurrent DOS Commands

DOS	CDOS	DOS	CDOS
APPEND	APPEND	MKDIR	MKDIR
ATTRIB	ATTRIB	***MODE***	***CCIMODE***
BACKUP	BACKUP	MORE	MORE
BREAK	BREAK	PATH	PATH
CHDIR	CHDIR	PRINT	PRINT
CHKDSK	CHKDSK	PROMPT	PROMPT
CLS	CLS	RECOVER	RECOVER
COMMAND	***CDOS***	RENAME	RENAME
COMP	COMP	REPLACE	REPLACE
COPY	COPY	RESTORE	RESTORE
DATE	DATE	RMDIR	RMDIR
DEL	DEL	SET	SET
DIR	DIR	SHARE	SHARE
DISKCOMP	DISKCOMP	SORT	SORT
DISKCOPY	DISKCOPY	SUBST	SUBST
EDIT	***EDITOR***	TIME	TIME
ERASE	ERASE	TREE	TREE
FDISK	FDISK	TYPE	TYPE
FIND	FIND	VER	VER
FORMAT	FORMAT	VERIFY	VERIFY
KEYB	KEYB	VOL	VOL
LABEL	LABEL	XCOPY	XCOPY

Additional CCI Concurrent DOS commands		
8087	CCIMAP	FLAGSET
&CHANGE	CCIMODE	FLAGWAIT
&CLEARCCI	CCITERM	FM
&CLRHIST	CCIUSER	FORCE
&CMDRESET	CFDISK	FUNCTION
&DISABLE	CHSET	HELP
&ENABLE	COMS	JOBMGR
&ENVRESET	CONFIG	KEYB
&SETCCI	COPYMENU	LIMSIZE
&SHLRESET	CREAD	LOADSYS
ADUMB	CSPOOL	LOGIN
AREAD	CWRITE	LOGOUT
AUX	DELQ	LOOKOUT
AWRITE	DOSVER	LWRITE
BANK	DSKMAINT	MEMSIZE
BWI	EDITMENU	MOVE
BYE	EMAIL	NOSWITCH
CCICOUNT	ERAQ	OPENS
CCIDUMP	FEATURE	PASS
CCIGEN	FIXCB86	PASSWORD

continued on next page

PDELAY	SECURITY	TRANSIT
PIFED	SETUP	USERNAME
PORTS	SHELL	WHATFLAG
PRINTER	SHOW	WHO
PRINTMGR	SID86	WHOAMI
REBOOT	STOP	WINDOW
REMOTE	SUSPEND	XATTRIB
RENDIR	SYSTAT	XDEL
RUNMENU	TAKEOVER	XDIR
SCHEDULE	TERMSET	
SCRIPT	TOUCH	

DOS vs. CCI Concurrent DOS BATCH commands

DOS	CDOS	DOS	CDOS
CALL	CALL	IF	IF
ECHO	ECHO	PAUSE	PAUSE
EXIT	EXIT	REM	REM
FOR	FOR	SHIFT	SHIFT
GOTO	GOTO		

Additional CCI Concurrent DOS BATCH commands

AND	INPUT	PAUSEERR
CASE	INPUTC	RETURN
DIREXIST	INPUTX	SWITCH
ECHOERR	ON	
GOSUB	OR	

Command Line Operation

The term command line refers to operations done at the system prompt (for example C:\>). All operating system commands can be used from the command line. The CCI Concurrent DOS command line also responds to special control key combinations. Control keys are typically used by simultaneously pressing the control key (usually labeled CTRL on the keyboard) and some other key. Control keys can also be used to edit the current command line or recall a previous command, manage video output, begin and end printing, switch to other sessions, etc.

Some of these functions involve manipulating the command line history buffers. The command line history buffer remembers the last few command lines. Previous command lines can be recalled and reviewed by pressing a control key (CTRL-E) or the UP-ARROW on the keyboard. All of the control codes and their corresponding functions are listed as follows.

Command Line Action Keys

CTRL-C	Stops the program that is currently running; also stops output to the printer following a PRINTSCRN.
CTRL-I	Same as the TAB key.
CTRL-O	Stops the video output to the foreground session. The program in that session continues to run, however. Video output resumes upon depression of any other key.
CTRL-P	Sends screen output to both the printer and the screen. A ^P appears in the session status line. Repeating CTRL-P cancels the printer output and removes ^P from the status line.
CTRL-Q	Restarts screen output that was halted with the CTRL-S combination.
CTRL-S	Stops screen output, pausing program operation.
CTRL-Break	Has the same effect as CTRL-C, stopping the program that is currently running or the output to the printer from a PRINTSCRN
PrintScrn	Copies the foreground session's screen to the printer. The TERMSET command can do a PRINTSCRN from the command line.
CTRL-1	Switches to session 1. Use the numeric keypad 1 key.
CTRL-2	Switches to session 2. Use the numeric keypad 2 key.

CTRL-3	Switches to session 3. Use the numeric keypad 3 key.
CTRL-4	Switches to session 4. Use the numeric keypad 4 key.
CTRL-5	Switches to session 5. Use the numeric keypad 5 key.
CTRL-6	Switches to session 6. Use the numeric keypad 6 key.
CTRL-7	Switches to session 7. Use the numeric keypad 7 key.
CTRL-8	Switches to session 8. Use the numeric keypad 8 key.

Command Line Editing Keys

Certain key combinations can be used to edit the command line entries.

CTRL-A	Move left one word
CTRL-B	Bounce between ends of the command line.
CTRL-D	Move right one character (or use the RIGHT-ARROW key)
CTRL-E	Move up in command history buffer (or use the UP-ARROW key)
CTRL-G	Delete character to right of cursor (or use the DEL key)
CTRL-H	Delete character to left of curser (or use the BACKSPACE key)
CTRL-J	Enter command line (or use the ENTER key)
CTRL-K	Delete to end of line
CTRL-M	Enter command line (or use the ENTER key)
CTRL-Q	Move to the beginning of the line (or use the HOME key)
CTRL-R	Turns history buffer search mode off/on for the current line. Search mode tries to find the command line in the history buffer that matches the characters currently on the command line.
CTRL-S	Move left one character (or use the LEFT-ARROW key)
CTRL-T	Delete word right
CTRL-U	Delete to beginning of line
CTRL-V	Toggle insert/overstrike mode on/off
CTRL-W	Move to the end of the line (or use the PGUP key)

CTRL-X	Move down in command history buffer (or use the DOWN-ARROW key)
CTRL-Y	Delete entire line
CTRL-\	Enter next character without interpretation. For example a CTRL-G normally deletes a character. CTRL-G is also the ASCII bell character. To insert a bell on the command line, type CTRL-\ CTRL-G.
CTRL-_	Toggles the default search mode on/off and sets the current line's mode to the new default. Search mode default is set to off.
END	Move left one word
PGDN	Move right one word

Function Keys

The CCI Concurrent DOS system supports the setting of up to 50 combinations of programmable function keys (PFK's) on the keyboard. PFK's can invoke any operating system command with just one or two keystrokes. The system console has a set of function keys for each session. All other PCTERM workstations share another single set of function keys. PFK's can be used from the command line; however, control of the function keys in any session is turned over to a DOS program when it is invoked on that session. When the CCI Concurrent DOS software is first installed, the function keys are set so that F1 runs the HELP utility and F3 recalls the last command line. All other function keys are blank.

To change the definitions of those keys, use the FUNCTION command. The F1 through F10 keys can have definitions of up to 20 characters. The remainder of the 50 programmable keys are obtained by combining the F1-F10 keys with the keys on the numeric keypad, the ALT key, the CTRL key and the SHIFT key. Each of those combinations can have definitions up to four characters long.

Redirection

The CCI Concurrent DOS operating system has the ability to take the output that would normally go to the screen and redirect it to another device, such as a file or printer. Redirection is accomplished using the symbols "<", ">", ">>", or "|" and the appropriate device or filename, with optional spaces between. For example, to direct the output of any command to a disk file, use the following syntax:

```
COMMAND > FILENAME
```

Likewise, using the ">" syntax and substituting a device name for the filename causes the output of the command to be routed directly to that device. For example:

```
DIR C:\CDOS > PRN
```

causes a listing of the files in the CDOS subdirectory to be printed to the PRN device.

Using the ">>" syntax takes the output and appends it to the end of the file specified, rather than creating a new file. Specifically, the command

```
TYPE MOREINFO.DAT >> INFO.DAT
```

takes the contents of MOREINFO.DAT and appends it to the end of INFO.DAT, which now contains the contents of both files.

Using the "<" syntax allows the input of command parameters from a batch or text file. For example, using

```
BACKUP < SAMPLE.TXT
```

causes a backup to be done based on the parameters specified in the SAMPLE.TXT file. This is one way to automate procedures.

Piping

The CCI Concurrent DOS system also can pipe the output of one program into another program's input. The command line syntax for piping uses "|" to separate the two programs. There are several commands where piping can be used to great advantage. They include the FIND, MORE and SORT commands. MORE displays system output a screen at a time, SORT sorts output alphabetically and FIND locates and pulls out certain items of data from the output. Combining these with other operating system commands through piping can increase their utility. For example, the DIR command's output can be piped through SORT to alphabetically sort the directory display.

```
DIR | SORT
```

TYPE's output could also be piped through MORE to display a file one screen at a time, making it easier to read.

```
TYPE TECHINFO.WS | MORE
```

On-Line Help

The operating system includes an extensive on-line help facility that is accessed by typing HELP at the command prompt, or by pressing the F1 key. More than 70 topics are available, with detailed instructions on using system facilities.

This page intentionally left blank.

Chapter 2:
Installation and Configuration

Installation and Configuration

Installation Checklist and Hardware Review

Review the following list since the successful installation and configuration of the multiuser operating system depends on the use of appropriate hardware and having the correct peripherals and tools. Details on each of the hardware components below are listed in the *CCI Concurrent DOS Applications and Hardware Guide*, under Technical Considerations.

Computer

- Does the computer have sufficient RAM memory to run the necessary applications?
- If this is a new model computer, has CCIDUMP been run to evaluate it?
- Note the number and type of ports the computer contains (for example, COM1, COM2, parallel ports, etc.)
- Does the computer have sufficient board slots and power to support the initial installation and expansion requirements?
- If a Dealer Starter Kit is available, has the Windsor Technologies PC-Technician™ diagnostics been run to verify the computer hardware?

Display monitor

- Is the auto-switch on the system EGA or VGA adapter set to OFF?

Hard disk

- Does the ROM BIOS support the hard drive?
- If the hard disk has more than 1024 cylinders, is a remapping controller in use?

Multiuser serial boards and display adapters

- Does this operating system include drivers for the boards or adapters being used?
- Has the board or adapter been configured as recommended in Chapter 2, Hardware Configuration, in the *CCI Concurrent DOS Applications and Hardware Guide* and installed in the computer?
- If VGA MaxStation™ is being used, has the VGA adapter in the computer been disabled or replaced with a compatible VGA adapter from Maxspeed?

Workstations

- Have all serial terminals been configured for PCTERM personalities?
- Note the number and type of ports the serial terminal(s) have (for example, main/modem/AUX/serial/parallel)

Tape drives

- Note the interrupt the tape drive uses. Under CCI Concurrent DOS, the tape backup must use a unique interrupt (IRQ) Other devices in the system cannot use the this IRQ.

Modem

- Has one of the modems recommended by Concurrent Controls been selected?
- Does the modem have a manual?
- Has a special cable for the modem been located and purchased?

Cabling

- Is the cabling shielded, with at least four wires (six wires for a modem)?
- Is the cabling properly grounded?
- Does the terminal cabling avoid close contact with fluorescent lights, power cables and other devices that cause electromagnetic interference?
- Are the appropriate hoods and adapters available for connecting the terminal ports to the serial port board, or serial ports in the computer (DB9 or DB25)?

Printers

- Is the printer set or switched for the proper mode, that is, serial or parallel?
- If a printer is being attached to a terminal, has the terminal's AUX port been configured for the appropriate handshaking (hardware and/or software)?

Mouse

- Is the mouse compatible with the supplied mouse driver (MOUSE.SYS)?
- Has the application(s) been configured for use with a bus mouse?

Tool Kit

There are a number of tools which can make the job of system integration much easier and ensure more reliable results. The recommended tool kit includes:

- RS232 Tester
- Breakout box
- 6-foot straight-through tested and proven cable.
- Gender changers
- Null modem adapters

The RS-232 tester displays the signals on the serial port to which it is attached. The RS-232 breakout box allows the changing of the individual pin-to-pin connections for testing purposes. Since the industry offers both modular, DB9 and DB25 connectors, always carry the necessary adapters

to work around connecting one to the other. All of the above equipment is contained in the Concurrent Controls Dealer Starter Kit, which is available at a nominal cost. Call 800-487-2243 for more details.

Pre-Installation Considerations

Using DOS

Before CCI Concurrent DOS is installed, decide whether DOS should also be installed, and, if so, what version of DOS should be used.

CCI Concurrent DOS does not require DOS to operate; it is a full replacement product for DOS. It can co-exist with DOS on the same hard disk if desired. If DOS is currently installed on the hard disk of the computer, CCI Concurrent DOS can be installed without altering the DOS files and configuration.

For installations that want or need DOS, one of the following is recommended:

- Compaq DOS 3.31
- MS-DOS 4.01 or higher
- DR-DOS 3.4 or higher

All of these support larger hard disk partitions (32MB) with the same sector size, which is needed to use the software disk caching included with CCI Concurrent DOS. If DOS is used, the DOS FDISK and FORMAT utilities can be used to create all partitions. If this has already been accomplished, move ahead to the Installation section of this chapter.

If DOS is not needed, and DOS is not currently installed on the computer's hard disk, all hard disk partitioning and formatting can be done with the FDISK and FORMAT utilities included in the CCI Concurrent DOS system.

Hard Disk Preparation

The CCI Concurrent DOS FDISK utility creates partitions that are DOS 4.X/5.X compatible. These partitions can be up to 740MB in size. To set up the hard disk, place disk 1 of the CCI Concurrent DOS delivery diskettes in drive A: of the computer and reboot. When the system boots from the floppy, it presents a menu of choices, including F1 for installation. Choosing F1 provides an option of continuing the installation, preparing the hard disk or requesting help.

The hard disk of the computer must be partitioned and formatted for installation of the operating system. Choose F3 to provide a function key driven menu that guides the user through the partitioning and formatting of the hard disks. At completion of these operations, the user is returned to the first installation menu. Select F10, installation, and proceed.

Installation

Installation of CCI Concurrent DOS requires the delivery disk set and this manual. The process is begun by placing disk 1 of the disk set into the A: drive and rebooting the system. When the system loads from the floppy, the CCI Concurrent DOS sign-on screen is presented, with options to install or exit.

Selecting the installation option provides the user with a function key driven menu for installing, preparing the hard disk or requesting help. If the hard disk is already partitioned and formatted, installation can proceed. The installation process checks for existing systems, and requests input for the drive and location of the system files it will be copying to the hard disk. When the contents of one floppy disk have been copied to the hard disk, the removal of that disk and the insertion of a new disk is requested.

The installation process creates a new directory called CDOS and copies into it system utilities, diagnostics and other programs from the delivery diskettes. Within the CDOS directory, three other subdirectories are created — SPOOL, SECURITY and MAIL. The print spooler files, system administration and file security files, and the electronic mail files are copied into their respective subdirectories and deleted from the CDOS subdirectory. Finally, other system files are then copied into the root directory.

Before installation is complete, one of two methods of loading the operating system must be selected. A menu offers F1 for Loader and F2 for Loadsys.

```
F1  Loader
F2  Loadsys
```

One method of loading the operating system is to use the LOADER.COM utility, which presents the user with a function key driven menu asking which operating system to load. This menu offers F1 for DOS and F2 for CCI Concurrent DOS.

```
F1  Standard Boot of bootable partition (DOS)
F2  CCI Concurrent DOS
```

Use of LOADER.COM prevents the loading of DOS prior to CCI Concurrent DOS and is recommended if the system is used, at times, under DOS or Windows. If MS-DOS 6.0 or greater will also be used, LOADER.COM must be used to avoid conflicts with high memory drivers. It can also be configured for automatic loading after a predetermined time. LOADER records a boot-track on the hard disk and requires a specific UNLOAD action to remove it. More information on LOADER.COM is provided in the *CCI Concurrent DOS Utilities and Troubleshooting Guide.*

The LOADSYS utility allows DOS to boot-up first and then allows the user to boot CCI Concurrent DOS. LOADSYS adds the line `LOADSYS ASK` to the

AUTOEXEC.BAT file. This results in a prompt to the user as AUTOEXEC runs asking for use of CCI Concurrent DOS or staying in DOS.

```
Load CCI Concurrent DOS? (Y/N)
```

To eliminate the prompt and boot the system automatically into CCI Concurrent DOS after DOS has been booted, delete the word ASK from the LOADSYS ASK line added to AUTOEXEC.BAT.

The LOADSYS INSTALL utility records details of the exact hardware used in the video, hard disk, and memory portions of the computer and saves that information to disk in a file called LOADSYS.COM, which is used by the CCI Concurrent DOS operating system.

After selecting the loading method, the installation process continues and completes. When finished, the system displays:

```
Hit any key to reboot the system.
```

When a key is depressed, the CCI Concurrent DOS copyright screens and detailed hardware overview are displayed on the screen, with the operating system prompt, (probably C:\>) below. Installation is now complete and configuration of the system software can begin.

Operating System Configuration

The basic configuration of the CCI Concurrent DOS operating system can be divided into several basic sections, including:

- Configuring the host computer and system console, including:
  - COM1 and COM2
  - system memory
  - system and session reboot capabilities
  - system features
- Configuring multiuser serial ports and/or multiuser display adapters.
- Customizing CCONFIG.SYS.
- Enabling disk caching in CCONFIG.SYS.
- Creating STARTxxy.BAT files.

After the basic system is configured, all or part of the extended capabilities of the CCI Concurrent DOS operating system can be implemented. These subsystems include:

- System Administration / Security
- Electronic Mail
- PC Terminal Emulation
- Remote Access and Support
- Scheduler
- Batch Command Processor Extensions

Configuring the Host Computer and System Console

1 Configuration of the host computer and system console, as well as reboot capabilities and system parameters, is done through the SETUP utility. SETUP must be invoked from the root directory, as it modifies the CCPM.SYS operating system file. It can be run from any console and is function key driven. At any time during the use of SETUP, pressing the F1 key provides help screens on the area of SETUP being accessed. Begin system configuration by entering the SETUP command at the command prompt and pressing the Enter key.

```
C:SETUP
```

Configuring COM1 and COM2

2 Choose F3 on the "Main" menu to setup COM1 or COM2 ports or Workstations. Choose F3 again and the "Setup COM1 and COM2" menu displays.

3 The F2 key displayed in the upper right column toggles between COM1 and COM2. F3 toggles between the type of device the COM port is configured for. As F3 changes, the other values on this screen change to reflect the recommended defaults for each parameter for the selected device type. In most cases, select the type of device under F3 and then check F4 to be sure the baud rate shown reflects the baud rate of the peripheral chosen.

Possible Function Key Settings

F3	Device type: Printer/AUX Port, Terminal, Mouse
F5	Parity: Even, Odd, or None
F7	Data bits: 7 or 8
F4	Baud rate: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 57600, or 115200.
F6	Stop bits: 1 or 2
F1	Provides online help
ESC	Returns to menu.

4 After setting the device type, press F8 to go to the Serial Port Protocol Menu. The exact protocols chosen depend on the hardware and cabling in use, but some guidelines apply:

Input Protocol (left column) is generally turned off, and some portion of Output Protocol turned on. Those choices are:

F4	DSR (Data Set Ready) is hardware handshaking for terminals.
F6	CTS (Clear to Send) is hardware handshaking for modems.
F8	XPC is software handshaking.
F9	DCD should not be turned on unless a modem (either internal or external) is attached to this port and the Data Carrier Detect line (pin 8) connected.

Use the PORTS utility on the COM port (with cable attached) to determine what protocols can and can not be used. If PORTS does not report ON for a signal, that choice on the Serial Port Protocol menu can not be used. After enabling the appropriate choices on the Serial Port Protocol menu, go to the next step.

5 Use the Esc key to exit from this menu and the prior menu to the Setup COM1/2 Workstations menu. If either COM1 or COM2 were enabled as Terminals, choose the F4 Setup COM1 and COM2 Workstation Features selection. Otherwise, go to step 7.

6 The menu displayed is Select COM1/2 Serial Console Features. This screen is preconfigured to reflect PC Terminal Emulation Enabled, Four Virtual Consoles, and Monochrome Terminal. To modify any of these (for example, to support the Relisys TR-170 terminal) depress the appropriate function key until the correct option is displayed. Possible choices are:

F4	PC Emulation ENABLED should always be set unless an ASCII terminal is to be used (for example, Wyse 30)
F7	1, 2, 3, or 4 Virtual Consoles
F8	Monochrome Terminal, Color CCI TERM, Color Wyse 370, Color Relisys TR-170

Once this screen reflects the desired characteristics for this workstation use the ESC key to exit to the previous menu.

Configuring System Memory

7 At the Setup Workstations menu, press ESC once to return to the main menu. The next area to consider is F4, System Memory. The CCI Concurrent DOS operating system is shipped with memory scanning already enabled. Modifications to the System Memory area should only be necessary if:

- the banking window is too small, or
- LIM needs to be enabled, or
- RAM disk needs to be enabled.

Press the CTRL key and the 2, 3, or 4 key on the numeric keypad to switch to another session and run the BWI (Banking Window Is) command. If BWI returns an amount of memory equal to or greater than the amount of memory needed by any of the applications on the system, then System Memory does not need to be altered.

8 Use the "System Memory" option (F4) to begin enabling LIM. On the Setup System Memory Menu, select F4 to enable LIM and F6 to select the LIM option. The TPA option is recommended, which makes LIM available to only those sessions that need it, without impacting the memory available to others on the system. Finally, use F8 to set the LIM memory per process — the default is 1024KB.

NOTE: *If the LIM TPA option is selected, the PIFED utility must be run on those programs that require LIM.*

To enable a RAM disk, select F4 on the main SETUP display for System Memory and then F3 for RAM disk. Press F3 again to specify the size of the RAM disk to be enabled. After entering that value, press ESC twice to return to the main menu.

NOTE: *In addition, the following line must be added to the CCONFIG.SYS file:*

```
DEVICE=C:\CDOS\VDISK.SYS /S /E
```

Setting System Console Sessions and System Reboot Capabilities

9 At the "Main" menu press F5 MISC: Setup Host Workstation. Use F2 to increase or decrease the number of sessions supported on the main console. Use the F5 selection to disable the CTRL-ALT-DEL sequence at the system console so that it does not reboot the entire system. Concurrent Controls recommends using this option to increase system and data integrity; it is not good to reboot while applications are running and files are open. Press F7 to enable CTRL-ALT-DEL by session,

providing session reboot capability, so individual sessions can be rebooted without affecting other work on the system.

Customizing System Features in SETUP

10 At the "Main" menu, choose F6 to modify the number of file handles available per session and systemwide, with defaults of 255 and 1024 respectively. Generally, these settings are adequate and do not need to be changed for testing purposes. Choose F7 to increase the number of automatic jobs that can be run by the system. System default is four jobs. The F7 screen also contains information on setting up the print spooler. Actual print spooler activation is done through batch files, see Chapter 3 for details. Finally, F8 on the main menu activates the security system (discussed in Chapter 4).

11 This completes the configuration using SETUP. At the main menu press F10, then F5 to save the changes made to the CCPM.SYS operating system file.

12 Reboot the system and test the devices connected to COM1 and COM2. If they do not respond correctly, refer to the *CCI Concurrent DOS Utilities and Troubleshooting Guide.* If the serial port parameters need to be altered, that can be done through the SETUP utility or through the CCIMODE utility at the command line. CCIMODE can modify any serial port parameter except for PORT TYPE. See the online HELP Utility for more details on CCIMODE. When the COM1 and COM2 devices have tested out correctly, go ahead to the next step.

Configuring Multiuser Serial Ports and Display Adapters

13 If the system being configured does not include a multiuser serial board or multiuser display adapter, move ahead to step 16. Otherwise, begin configuration of those boards by invoking the CONFIG utility in the root directory. The main CONFIG screen reflects the type of internal serial driver in the operating system. If multiple serial drivers are present, then CONFIG displays a menu listing each driver. The user highlights the driver for the board to be configured, and the main CONFIG screen for that driver is displayed. Highlighting either and depressing the ENTER key results in a CONFIG display specific to your choice.

CONFIG is a single-screen utility similar to a spreadsheet that offers interactive help on all parameters, displaying all possible choices for each, and checking entries for accuracy and potential conflicts. CONFIG supports the use of both the arrow keys and of the WordStar® control keys for moving around the screen. Moving the cursor to a new

field causes HELP information to be displayed in the HELP window. The selected parameter field's function is described and available options are listed. In addition, a special note may be displayed about the selected field. Whenever an option is changed, the new option is highlighted. Scrolling through all the options wraps back to the original option, which will not be highlighted.

CONFIG can be automatically configured by using the FORCE command in conjunction with it. All of the keystrokes necessary to modify the CONFIG parameters can be passed by FORCE to CONFIG and the necessary FORCE entries placed in a batch file.

CONFIG may also be used to immediately change and reinitialize an add-on port's baud rate, protocol, etc., with the (I)nitialize option. It can also be used to clear all characters from a specified serial channel. The only parameter change that requires rebooting to activate is PORT TYPE.

14 Set the CONFIG parameters to match those reflected in the Hardware Configuration chapter of the *CCI Concurrent DOS Applications Guide* for the particular type of board to be used. If a standard multiuser serial board is in use, be particularly careful to set the parameters on the far right side of the CONFIG screen accurately. They differ from board to board and are the way in which CONFIG can be customized to support a particular standard board.

Multiuser Display Adapters — VGA MaxStation

15 If VGA MaxStation is being used, please review the VGA MaxStation section in the *CCI Concurrent DOS Applications and Hardware Guide.* Follow the instructions for installing the MaxStation hardware and configuring the ports on it with the CONFIG utility.

Device Setup

16 At this point, the operating system knows the type of devices that is in use. Now configure the particular device type (terminal, printer, modem) and then test it according to the steps listed on one of the four next pages:

- Terminal
- PC as terminal
- Modem
- Printer

Handshaking

When selecting the handshaking to use for terminals, printers or other devices, either hardware and/or software handshaking can be chosen. Determine the best chose after reading the following paragraphs.

Software handshaking can be either XON/XOFF or XPC. XPC is a special version of XON/XOFF designed for PC terminals. It does not use the same STOP and START codes, since the XON/XOFF stop and start codes (CTRL-S/CTRL-Q) conflict with the PC TERM definition. XPC is used with PC terminals, XON/XOFF with printers, ASCII terminals and other devices.

Hardware handshaking is typically DTR/DSR on standard port boards, and is defined differently by each intelligent board. For example, the Arnet Smartport uses CTS, and the Digiboard COMxi uses DTR and CTS. The type of handshaking used depends first on the type of cabling and board being using. If only 3-wire cabling is in use, only software handshaking can be used (even this is problematical, as CCI Concurrent DOS sends out data so quickly that it may overrun a 3-wire cable — 4-wire cable is the minimum recommended).

If the cabling used has at least four wires, either hardware or software handshaking, or both can be used. Using software handshaking with a very long cable may result in noise or signal reflections causing an XPC OFF character to be sent. With XPC handshaking, that would be a 65, which can be seen in the PORTS utility. The issuance of this character can cause the terminal or device attached to that port to appear to be locked up, as it waits for the XPC ON character.

On the other hand, if only hardware handshaking is selected, and a cable has a bad pin or wire, it is very difficult to diagnose without a breakout box.

If a breakout box is available, the use of hardware handshaking is recommended as the minimum. If a breakout box is not available, then the use of software handshaking only is recommended, on a temporary basis, until a breakout box can be acquired. If a modem is in use, RTS/CTS handshaking should be set on both the receive and transmit side (both in the left and the right columns of the SETUP utility).

NOTE: *A breakout box is included in the Dealer Starter Kit that can be obtained from Concurrent Controls.*

Terminal Setup

16a Set the parameters in the terminal's setup screen to match the parameters in SETUP (for COM1 & 2) or in CONFIG (for all add-on ports). Be careful to match baud rates, handshaking, parity, etc. In order to have multiple sessions (multitasking) and/or run DOS programs, the termi-

nal's personality must be set to PCTERM, and the terminal keyboard set to SCAN. Confirm that the operating system is configured for a MONO or COLOR terminal (done in SETUP and CONFIG) by using the TERMSET utility to display the current settings for a particular workstation or user number.

After the terminal is configured, connect the appropriate cables and test the workstation. For details on the recommended wiring for cables and workstations refer to the Hardware Configuration chapter of the *CCI Concurrent DOS Applications and Hardware Guide.* Is there a prompt on the workstation? If not, use the PORTS diagnostic (see online HELP utility for more details on using ports) to confirm that the physical connection exists.

PC/AT Terminal Setup

16b If a PC, XT, or AT is being used as a local workstation, review the *CCI TERM Terminal Emulator Users Guide,* which covers installation and use of CCI TERM, the Wyse 160/370 terminal emulation package bundled with this operating system. For workstations attached to COM1 or COM2, CCI TERM should already have been selected as the workstation type in SETUP. For workstations connected to all other ports, use the TERMSET utility to define the port that the workstation is attached to for CCI TERM, as follows:

```
TERMSET Ux CCITERM CCPM
```

Confirm that the CCI TERM configuration parameters match those recommended for local workstations; in particular, check that the Shift-F10 parameter (WAIT FOR) is set to NO.

Once the workstation has been configured, connect the appropriate cables and test the workstation. For details on the recommended wiring for cables (both for workstations and other peripherals) refer to the Hardware Configuration chapter of the *CCI Concurrent DOS Applications and Hardware Guide.* A null modem adapter is needed between a 25-pin serial port on the host computer and a 9-pin serial port on the computer running CCI TERM.

Modem Setup

16c If this modem is to be used for dial-in access, review Chapter 5, Remote Access before proceeding. Follow the step-by-step instructions in that chapter for configuring it.

Confirm that the serial port to which the modem is attached is defined to reflect the use of that port. For example, if this modem is used for dial-out communications, the serial port type should be AUX in SETUP or CONFIG. If the port is used for dial-in access, and, therefore,

has a computer or terminal at the other end of it, the port type should be PCTERM terminal (PTxx) in CONFIG.

Set DCD active on the serial port being used by the modem, using the SETUP utility for COM1 and COM2, and CCIMODE for all add-on serial ports. In SETUP, DCD can be found on the Serial Protocol screen under (F9); in CCIMODE use the syntax `CCIMODE Px DCD CCPM`.

Printer Setup

16d Connect all printers to the appropriate serial ports. Test those printers from the operating system first — if they do not work there, they will not work within an application. One of the easiest tests is to press CTRL-P and watch for the `^P` to appear on the console status line. After `^P` is displayed, pressing the Enter key on the keyboard should cause a prompt symbol to be printed on the printer. Press CTRL-P again to stop printing. For laser printers, use the command LWRITE 12 to force a formfeed. In addition to printing with ^P, printing can also be tested using the following command, which prints the AUTOEXEC file to the default printer:

```
COPY \AUTOEXEC.BAT PRN
```

Use the PRINTMAP /P command to display a list of valid printer numbers and then use the PRINTER command to change the printer number assigned to the current session.

Mouse Setup

16e Configure the port with SETUP or CONFIG to be a MOUSE. Set the baud rate to 1200, 8 data bits, 1 stop bit, no parity and disable all handshaking for the port. Newer mice may require a baud rate of 2400 baud; check the mouse documentation.

Use the TERMSET utility to assign a mouse to a particular workstation, as follows:

```
TERMSET MOUSE AUX# CCPM
```

AUX# represents the number of the serial port that the mouse is attached to. TERMSET must be run on the workstation that will be using the mouse.

Include the mouse driver line in CCONFIG.SYS:

```
DEVICE=C:\CDOS\MOUSE.SYS
```

Connect the mouse to the proper COM or AUX port and reboot.

Confirm that MOUSE.SYS is in the CDOS subdirectory. REBOOT and check the hardware overview to verify the mouse driver is loading and (if using COM1 or COM2) a mouse is logically attached to the port.

Configure each application for a bus mouse, or for a mouse type that uses a device driver. Most applications assume they can directly access the hardware if they are configured for a serial mouse. Configuring for a bus mouse results in the application using the device driver to access the mouse.

Troubleshooting Mouse Problems

COM1: Run the ADUMB program and move the mouse. The screen should show something. Since AUX 0 corresponds to COM1 and printer 3, enter `PRINTER=3` to confirm the serial port is not being used by some other task. The response should indicate that printer 3 is currently available.

COM2: Run the ADUMB program and move the mouse. The screen should show something. Since AUX 1 corresponds to COM2 and printer 4, enter `PRINTER=4` to confirm the serial port is not being used by some other task. The response should indicate that printer 4 is currently available.

If the mouse activity is uncontrolled and erratic, check and modify the baud rate using CCIMODE.

Finished Setup

All appropriate parameters for workstations, printers, and other devices have now been set and saved in the operating system.

If all workstations and peripherals have been tested and are operating correctly, move on to customizing the system. If the suggestions contained in these sections have been tried and have not resolved the problems, contact Concurrent Controls Technical Support at 415-873-6240.

Customizing the System

17 This step in configuring the CCI Concurrent DOS system sets up the CCONFIG.SYS file. This file is analogous to the DOS CONFIG.SYS file and contains parameters for buffers, fastopen, aligned memory, and device drivers. The CCONFIG.SYS is located in the root directory of the drive from which the operating system boots. The parameters in the CCONFIG.SYS are read each time the system boots. If a change is made to the CCONFIG.SYS file, the system must be rebooted in order for the change to take effect.

Included with the operating system is a basic CCONFIG.SYS file containing the following:

```
BUFFERS=256
FASTOPEN=1024            FLAGS=254     QUEUES=10,100
;DEV=\CDOS\CACHE.EXE /S=1024
```

This file can be modified using the CCI Concurrent DOS EDITOR or using another text editor or word processor.

One of the most important CCONFIG options is the driver for activating software disk caching. This should always be activated unless a hardware caching controller is in use. To activate the cache, use a text editor to remove the semicolon (;). Then set the /S option to the size in KB of cache to be activated. It is recommend that at least 256K of cache for good performance and saving wear and tear on the hard disk. To determine what size cache provides optimum performance, use the CACHE utility after completing the system configuration. For more information on CACHE, see Chapter 2, Utilities Reference, in the *CCI Concurrent DOS Utilities and Troubleshooting Guide.*

Other CCONFIG.SYS options that may be needed are:

- DEVICE=\CDOS\VDISK.SYS/S/E (for using a RAMDISK)
- DEVICE=\CDOS\MOUSE.SYS (for using a mouse)
- BUFFERS=200 (increase disk buffers)
- LASTDRIVE=X (If the SUBST command is used, use this to indicate the last available drive in the system.)
- COUNTRY=X (If an international version of CCI Concurrent DOS is used, this parameter selects a locale.)
- INSTALL=\CDOS\KEYB.COMX (If an international keyboard is required.)

Additional details on the CCONFIG.SYS file and each of the options can be found in the *CCI Concurrent DOS Utilities and Troubleshooting Guide.*

18 This step in configuring the system creates and modifies batch files that configure and set each user's environment. Creating and editing these files is discussed in detail in Chapter 3. Batch files are used to set a user's environment, including path, printer, prompt, etc.; to invoke name service and/or security; and to execute commands and start programs. They can be set up to configure the system in any and all of the following ways:

- globally
- by physical workstation
- by user name
- by time of execution

Specifically, STARTxxy.BAT files can be used to configure physical workstations and are active in all systems. NAME.BAT files are used to configure for specific users and can only be used in systems using name service or with security active; using security is recommended. START000.BAT is the global START file. LOGIN.BAT is the global NAME file. CCIBOOT.BAT is a special batch file executed immediately after a system boots before any other activities are performed. It is only active in systems with CCI Security enabled (see the CONTROL file DO-UPON-BOOT option in Chapter 4, Security Subsystem).

After the system boots, CCIBOOT.BAT is the first batch file executed in secure systems. Otherwise, the STARTxxy.BAT files are run for each virtual console on the system. The xx represents the workstation or physical console number; y represents the session on that workstation. If a specific STARTxxy.BAT does not exist for a console/session, then START000.BAT is run for that console.

If Name Service and Security are used, a NAME.BAT file is automatically invoked whenever that person logs in, regardless of the location he or she logs in from. If there is no NAME.BAT file for that person, then LOGIN.BAT runs at that console. In a secure system, START files can be run before or after the login prompt is presented, or not at all. For details on activating and using all or part of security, see Chapter 4, Security Subsystem.

There is a START000.BAT supplied with the operating system. Use the EDITOR utility — a full-screen ASCII text editor — to modify it and make copies of it for specific consoles. For details on creating and using START and NAME batch files, review Chapter 3, Batch Processing.

19 Installation and configuration of the operating system is now complete. The CCONFIG.SYS and appropriate batch files have been customized for this installation. All peripherals (terminals, printers, etc.) have now been tested and found working. For any remaining problems, call Concurrent Controls Technical Support at 415-873-6240 or dial into our BBS at 415-873-6256.

Other subsystems may now be activated, configured and used, including:

- Security Subsystem — Chapter 4
- Remote Access and Support — Chapter 5
- Scheduler System — Chapter 6
- Menuing System — Chapter 7
- Print Spooler System— Chapter 8
- Novell NetWare Connectivity — *CCI Concurrent DOS Applications and Hardware Guide*
- Electronic Mail — *eMAIL™ Electronic Mail Users Guide*

Changing or Adding Hardware

When changes are made to the video, disk, memory, or BIOS portions of the computer hardware on which CCI Concurrent DOS is installed, the operating system must be updated to reflect those changes if the LOADSYS utility is used to load the software. If LOADER is in use, no changes to the system need to be made.

This is because LOADSYS stores a "picture" of those portions of the computer. That picture includes information on the video adapter, hard disk and controller, floppy disk and controllers, and motherboard; also, memory and any shadowing options in the BIOS.

To update LOADSYS after making hardware changes, boot from the original Startup Disk, abort the Install with F9, and then log into drive C:. Then type LOADSYS INSTALL and press the Enter key. The LOADSYS program now has an updated picture. This procedure is not necessary if LOADER.COM, an alternative operating system loader, is in use. For information on LOADER, refer to the on-line HELP system.

In addition, if the high memory map is altered (by putting in or taking out a device which has a footprint) the operating system needs to again scan it. Do this within the High Memory options in the SETUP utility (F4, F10). Set all memory blocks to AUTOSCAN, save parameters and reboot. This is done automatically when CCI Concurrent DOS is first installed.

Optimizing for Best Performance

Concurrent Controls operating systems have been engineered to support fast performance at all stations and devices. In order to get the best performance from the system, follow the guidelines included here.

All stations and peripherals should be run at the highest speeds possible. All terminals and workstations should be run at their maximum supported speed and remote workstations should use 9600 baud modems for optimal performance. Most terminals support at least 38.4 K baud; several (for example, Wyse 160 and 370) support 115 and 56 K baud respectively. Existing PCs that are running with CCI TERM should be using from 38.4 to 115Kb, depending on handshaking and cabling used. PCs and XTs, particularly older ones, may not be able to keep up with speeds greater than 9600 to 19.2Kb. While CCI Concurrent DOS supports workstations at slower speeds (some terminals only run at 9600 or 19.2K) overall best performance is received when all stations are running at faster speeds.

Increasing the BUFFERS= line in CONFIG.SYS to 512 generally improves performance and usually does not impact the banked window. The BUFFERS statement creates memory buffers used for holding directory entries. Each buffer required 512 bytes, so BUFFERS=256 uses 128 K of extended RAM. On systems with many files, increasing BUFFERS above

512 (so that all directory entries are stored) improves performance but must be balanced against the extra time that FILE FLUSH spends searching the additional buffers for changes.

FASTOPEN, another CCONFIG option, defaults with no entry to 8192K and can not be reduced. It sets aside memory to keep a list of the most frequently opened files and their location on disk. This allows the operating system to locate files quickly, without having to search the directory.

Use disk caching. All CCI Concurrent DOS systems include the hard disk caching driver CACHE.EXE. Using any cache at all improves the performance. The maximum size cache that can be created depends upon the layout of the physical hard disk being used and can be as much as 16MB. Since extended RAM is used for the software cache, plan on using additional RAM equal to the size of the cache. The cache driver needs only a small amount of memory in the high memory region, typically two 4K blocks.

The amount of cache optimal for the configuration depends on the hardware in use, and the applications being executed. The degree to which cache helps depends upon the size of the cache, the size of the programs and databases, and how fast the CPU can use it. The faster the processor, the more cache helps performance. The effect of different sizes of cache on the system can be evaluated with the CACHE utility, which reports back a hit/miss ratio on the cache. The higher the hit percentage, the better the performance received. The use of at least 256K and preferably 512K of cache is recommended in every installation, unless hardware caching is being used. If the system includes a hardware caching controller, the CACHE.EXE software caching should be disabled or reduced to a minimum (32K). Test the cache with real applications to evaluate.

Defragment the hard disk on a regular basis. A fragmented hard disk slows down the system and tape backups.. There are several DOS tools used to defragment the hard disk, including PC Tools and Norton Utilities. In general, it is recommended that these be run under DOS. If run under CCI Concurrent DOS, use the system administration's SINGLE mode (see the on-line help system for information of SINGLE and the CONTROL file) so that only the system console is active.

Chapter 3:
Batch File Processing

Batch File Processing

Batch files are ASCII text files with a .BAT file extension. Batch files contain a group of commands, one command per line, which are executed sequentially in a "batch". There are several distinct types of batch files, each of which is treated differently by the CCI Concurrent DOS operating system, but all of which can be executed by typing their name at the operating system prompt and hitting return.

Types of Batch Files

Batch files can be divided into three categories:

- Those that are executed at boot time — CCIBOOT.BAT and STARTxxy.BAT.
- Those that are executed automatically at login time — NAME.BAT and LOGIN.BAT (these require name service).
- Those that are executed from the command line — all others.

The startup and login batch files are run with command line parameters matching the CCI Concurrent DOS environment variable set used by &SETCCI. The first command line parameter is the current user number, the second is the workstation number, the third is the session number, the fourth is the user's name, etc. See the &SETCCI command in the *CCI Concurrent DOS Utilities and Troubleshooting Guide* for a complete list.

CCIBOOT Batch File

The CCIBOOT.BAT file in the SECURITY directory is a unique batch file that is executed immediately after the system boots and before any other STARTxxy.BAT batch files. CCIBOOT.BAT is only run if the CONTROL file in the SECURITY contains the line `DO-UPON-BOOT Un`, where n is the user number where CCIBOOT runs.

If both conditions are true, CCIBOOT.BAT executes on user console number n. CCIBOOT.BAT should be used for global operations that need to be performed only once each time the system is booted; for example, starting the printer spooler or invoking a file server.

STARTxxy Batch Files

START files are used to configure workstations, in contrast to NAME.BAT files for specific users. The "xxy" identifies the particular console that the START file applies to; "xx" represents the workstation (or physical console) number, and "y" represents the session number on that workstation. For example, the terminal connected to the COM1 port is workstation number 01 with session numbers 1 through 4. The COM1 terminal would use

START011, START012, START013 and START014 .BAT files. The WHOAMI utility will return the exact START file numbering for the console at which it is executed.

When the system boots, the CCIBOOT.BAT file is run, then the START files are run. Choose not to execute the START files at this time by using the CONTROL file START-FIRST EXCLUDE ALL option.

(See the CCI Concurrent DOS Users Guide, Chapter 4, Security Subsystem for more details) If a STARTxxy.BAT file does not exist for a specific console, then the global START000.BAT will be executed for that console. A separate STARTxxy.BAT file does NOT have to be made for every console. "Generic" start commands can be included in the START000.BAT file and specific STARTxxy.BAT files created only for those consoles which require special handling.

Alternatively, use the FORCE command to configure and run operations at multiple consoles from a single START file.

NAME Batch Files

NAME files, where "NAME" is the user's login name, are always used in conjunction with a login prompt from name service or the security system. NAME.BAT files are used for configuration and operations for specific users; they don't apply to any particular workstation and will be executed on any workstation the user logs into. NAMEy.BAT may be used, where "y" is the session number (1 through 4) on the workstation. If the security system is active, specify in the CONTROL file whether START files should be run on each workstation before anyone logs into it. In the PASS utility, specify whether or not the START files for a workstation should be run after each particular user logs in and also whether or not the user's NAME files should be run.

LOGIN Batch Files

LOGIN.BAT is a global batch file that is run after a user logs in if there is no NAME.BAT file for that user.

Other Batch Files

The uses to which batch files can be put are limited only by our imagination. Using the scheduler, you can execute batch files at a specific date and time. Batch files can also be interactive, requesting and waiting for user input. The remainder of this chapter covers the capabilities and functions available for use in BATCH files.

Creating a Batch File

Batch files are created using an ASCII text editor like the EDITOR program.

It is also possible to use the COPY CON command to copy input from the console keyboard directly to a file on disk. For example, one would create the batch file SAMPLE.BAT using:

```
C>COPY CON SAMPLE.BAT
COPY *.DAT C:\BACKUP
CD \BACKUP
DIR /W
```

Terminate the COPY CON function by pressing CTRL-Z and Enter. If SAMPLE is typed at the command prompt, the three commands in the SAMPLE.BAT batch file are executed.

There are several commands that are very useful within BATCH files. These commands are:

- FLAGWAIT and FLAGSET: Used to synchronize events using system flags.
- PDELAY: Used to delay the execution of the BATCH file by a number of seconds.
- FORCE: Used to configure and/or execute commands on multiple consoles from a single batch file; it forces the keystrokes specified directly into the keyboard buffer of the specified console.

Additional detail on using these commands can be found in the *CCI Concurrent DOS Utilities and Troubleshooting Guide.*

Running Batch Files

A batch file is normally run by entering the name of the batch file without the .BAT extension at the command prompt. It may be useful to specify the .BAT extension in case there is a .COM, .EXE, or .CMD file with the same filename as the batch file in the file search path. For example, if SAMPLE is run, and your current directory contains SAMPLE.EXE and SAMPLE.BAT files, the SAMPLE.EXE file is executed. If SAMPLE.BAT is run, it forces the execution of the SAMPLE.BAT file.

A batch file can be interrupt or terminated during processing by pressing CTRL-C. The question `Halt Batch Process (Y/N) ?` is then asked. A "Yes" answer terminates batch processing, a "No" answer continues processing commands in the batch file.

Environment

Every session on a CCI Concurrent DOS system has its own unique environment that contains information about the session. The environment can

be queried in a batch file to check for certain conditions. It is initialized when the system first boots and can be reviewed by typing "SET" at the command prompt. The typical default environment includes user, station and window number and the operating system executing. Additional variables can be added to it with the command SET variable = value. For example, the following lines add three variables - TPATH, PROD and LASER - to the environment.

```
set TPATH=F:\TEAMUP
set PROD=F:\PROD
set LASER=0
```

The &SETCCI command, when executed, adds 16 variables to a session's environment. These can be used to test for whether a batch file is running on the foreground or background, on the main console or a terminal, for example.

&SETCCI and &CLEARCCI

Syntax: &SETCCI

&CLEARCCI

The CCI extended command pair &SETCCI and &CLEARCCI set and clear CCI-specific environment variables for use in batch files. The CCI variables are:

CCI01 — User number: 3 digits. Range: 001-(number of sessions, or virtual consoles, on the system). The host is user 001 to 004, the first terminal begins with user 005, etc.

- *Example: CCI01=001*

CCI02 — Workstation number: 3 digits. Range: 00-(number of workstations, or physical consoles, on the system). The host is station 00, first terminal is station 01, etc.

- *Example: CCI02=01*

CCI03 — Session number: 1 digit. Range: 1-4. This parameter indicates whether you on the first, second, third or fourth session on the current workstation.

- *Example: CCI03=1*

CCI04 — User name: 1 to 8 characters. This parameter is taken from the %USERNAME% environment variable as set by the LOGIN or USERNAME commands. If name service or security is not enabled, the default user name is "NONAME".

■ *Example: CCI04=GREG*

CCI05 — Workstation printer number (i.e. the printer number assigned to the printer physically attached to your PCTERM terminal): 1 to 2 digits. Range: 0 or 3-(last PCTERM terminal on system). This parameter is 0 if you are running on the host console or an ASCII terminal; otherwise, a PCTERM terminal on COM1 is workstation printer 3, COM2 is printer 4, etc. Use PRINTMAP /P for a list of valid printer numbers for your system.

■ *Example: CCI05=5*

CCI06 — Default printer number (i.e. the printer number assigned to the current session): 1 to 2 digits. Range: 0-(last printer on system). LPT1 is printer 0, LPT2 is printer 1, LPT3 is printer 2, COM1 is printer 3, etc. Use PRINTMAP /P for a list of valid printer numbers for your system.

■ *Example: CCI06=12.*

CCI07— Default aux port number (a.k.a. communications port): 1 to 2 digits. Range: 0-(last aux port on system). COM1 is aux 0, COM2 is aux 1, the first add-on port is aux 2, etc. Use PRINTMAP /A for a list of valid aux port numbers for your system.

■ *Example: CCI07=6*

CCI08 — CCI console driver number: 1-2 digits. Range: see following table.

CCI09 — CCI console driver name: 1-25 characters. Range: see following table.

CCI10 — CCI driver port number: 1-2 digits. Range: see following table.

CCI11 — CCI driver port name: 2-4 characters. Range: see following table.

CCI08	CCI09	CCI10	CCI11
0	(reserved)		
1	HOST	1	HOST
2	LPT	1-3	LPT1-LPT3
3	COM	1-2	COM1-COM2
4	(reserved)		
5	STANDARD-8250	1-64	P1-P64
6	ARNET-SMARTPORT	1-64	A1-A64
7	WYSE-995	1-16	W1-W16
8	DIGIBOARD-Com-Xi 1-16	1-16	DC1-DC16
9	VIEWPORT-MONO	1-8	SS1-SS8
10	VIEWPORT-SERIAL	1-8	SP1-SP8
11	COMPUTONE-INTELLI-AT	1-16	CA1-CA16
12	DIGIBOARD-PC/Xx	1-64	DP1-DP64
13	(reserved)		
14	(reserved)		
15	MAXPEED-SSx	1-64	MX1-MX64
16	COMPUTONE-INTELLI-II	1-64	CI1-CI64
17	MAXSTATION-MONO	1-16	MM1-MM16
18	MAXSTATION-SERIAL	1-16	MS1-MS16
19	MAXSTATION-PARALLEL	1-16	MP1-MP16

CCI12 — Indicates the session is either the foreground or background session: 2 characters. Range: FG or BG.

■ *Example: CCI12=FG*

CCI13 — Video mode 1: 3 characters. Range: see below. This parameter expands upon the CCI14 parameter.

CCI13	Video mode
MDA	monochrome display adapter
HGC	Hercules graphics mono display adapter
CGA	CGA adapter
EGA	EGA adapter
VGA	VGA or PS/2 MCGA adapter

- *Example: CCI13=VGA*

CCI14 — Video mode 2: 4 to 5 character. Range: MONO or COLOR.

- *Example: CCI14=MONO*

CCI15 — Terminal type: 2 characters. Range: see following table

CCI15	Terminal type
HS	Host
PC	PC Terminal
SS	SmartScreen
TY	ASCII terminal, i.e. not PCTERM

- *Example: CCI15=HS*

CCI16 — LOOKOUT port type. Range: LOCAL or REMOTE.

- *Example: CCI16=LOCAL*

The CCI01 through CCI16 environment variables are also passed to the STARTxxy, NAMEx and LOGIN.BAT startup batch files as command line parameters. The arguments are accessed in the batch files using the standard "%N" and SHIFT batch file commands.

CAUTION: *The values of the CCI variables are valid at the time the &SETCCI was executed. If you run USERNAME, PRINTER or AUX to change your username, default printer, or default aux port, run &SETCCI again to update the environment.*

Batch File Variables

Batch files can access two types of variables: command line parameters and environment variables. The batch file filename and the command line parameters are accessed within a batch file by using the format: %N, where N is a number from zero to nine. When the batch file is run, %0 is the batch file filename, %1 is the first command line parameter, %2 is the second parameter, etc. For example, if SAMPLE.BAT includes:

```
COPY %1 C:\BACKUP
COPY %2 C:\BACKUP
```

and it is invoked with:

```
C>SAMPLE *.TXT *.CHP
```

the batch processor replaces %0 with the batch file filename SAMPLE.BAT, replaces %1 with the first command line parameter *.TXT and replaces %2 with the second parameter *.CHP and executes:

```
COPY *.TXT C:\BACKUP
COPY *.CHP C:\BACKUP
```

Up to ten command line variables can be accessed at a time in each batch file, named %0 through %9. Use the SHIFT statement when executing a batch file with more than 10 command line parameters.

Batch files can also test and use environment variables established with the SET or &SETCCI commands (see &SETCCI in chapter 2 of the *CCI Concurrent DOS Utilities and Troubleshooting Guide*). Environment variables are accessed within a batch file by using the format: %NAME%, where NAME is the environment variable's name. To use the environment variable OS in an IF statement, for example:

```
IF "%OS%"=="" GOTO IS_DOS
IF "%OS%"=="CDOS386" GOTO IS_CDOS
```

In this example, the batch file processor replaces "%OS%" with the value of the OS environment variable and then takes action based on that value. If OS has not been SET (i.e. %OS% is empty), execution continues at the "IS_DOS" label in the batch file. If OS is set to CDOS386, execution proceeds to the "IS_CDOS" label. This is a common technique when writing batch files which will be run under both DOS and CCI Concurrent DOS to execute commands specific to one operating system or the other. The IF and GOTO statements are described in more detail below.

The CCI Concurrent DOS command shell also sets an environment variable called ERRORLVL to the value of the exit code returned by the last .COM or .EXE program executed. This ERRORLVL environment variable gives greater flexibility when handling the DOS ERRORLEVEL. For example, the ERRORLVL environment variable can be saved to a log file to help track when a command fails:

```
BACKUP C:\ A:
ECHO Backup exit = %ERRORLVL% >BACKUP.LOG
```

Use the ERRORLVL environment variable with the CCI Concurrent DOS command shell's string comparison operators to check for a range of ERRORLEVEL's. For example, if an .EXE program defined a range of errors, where error numbers 1-10 were I/O errors and error numbers 11-20 were processing errors, the following example causes a conditional branch depending on the ERRORLVL variable.

```
IF "%ERRORLVL%" LE "10" GOTO IO_ERR
IF "%ERRORLVL%" LE "20" GOTO PROC_ERR
```

Execution of the batch file branches to the IO_ERR and PROC_ERR labels depending upon the value of the environment variable ERRORLVL.

The command shell's built-in TIME and DATE commands can also set and manipulate environment variables for use in batch files. See chapter 2 of the *CCI Concurrent DOS Utilities and Troubleshooting Guide*, for more details on the TIME and DATE /B and /E options.

Example: The following two batch and one script file can be used to accomplish the polling of multiple sites and downloading of data from each, at a specified time and frequency.

The first file DNLOAD.BAT is only two lines long and schedules the execution of the second file, every Tuesday and Sunday beginning at 1:00AM and completing no later than 6:00AM

```
rem  DNLOAD.BAT file
schedule connect.bat on 1-? every tues 0100 retries 300
schedule connect.bat on 1-? every SUN 0100 retries 300
```

The second batch file executes the TRANSIT script which dials the modem, connects and downloads the file. It also creates an activity log for each location polled, and concatenates those logs into an overall report.

```
rem CONNECT.BAT file
del transit.log  null
for %%p in (6976 6971 6982 6991 6951) do transit @cciscript
sysop GCDNLOAD
        ymodem %%p.txt %%p
copy transit.log dnload.rpt
force 01 type dnload.rpt | more
```

The last file is the TRANSIT script file, CCISCRIP, that is executed by the previous file, and performs the actual communications. It receives the variables passed to it by the CONNECT.BAT batch file and uses them to dial, login with name and password, and perform the file transfer using the specified parameters.

```
; CCISCRIP TRANSIT script file
;%1 = user name
;%2 = user password
;%3 = protocol
;%4 = filename
;%5 = phone number

IF CCI AUX 1
IF DOS COM 2
;use COM2 with DOS, AUX1 with Concurrent
LOG APPEND
;append the activity detail to the existing TRANSIT.LOG
INITMODEM
;initialize the modem using the standard escape sequences
;for Hayes compatible modems.
DIAL %5
;dial the number passed by the CONNECT.BAT batch file
WAIT 10 CONNECT
;wait ten seconds to connect
IF ERROR QUIT
;if no connect, exit the script and return to the batch file
PUT PCTERM
PUT 2CR
;if connect, put the channel into PCTERM mode and enter two
;returns as expected by a channel that is secure.
PUT %1
PUT %2
WAIT 15 "C:\"
;enter the name and password passed by CONNECT.BAT and wait
15
;seconds for a C prompt to be displayed.
IF ERROR DISC
IF ERROR QUIT
;if a C prompt isn't displayed, hang up the phone line,
exit the
;script and return to the batch file
PUT TRANSIT %3 SEND %4
;at the C prompt, enter the TRANSIT command for sending a
file,
;using the protocol and filename information passed by
CONNECT.BAT
%3
OVERWRITE
RECEIVE %4
;specify the protocol and filename to be received on the
local
;system and overwrite that filename if it already exists.
```

```
IF ERROR DISC
IF ERROR QUIT
;if that can't be done, hang up the phone line, exit the
script and ;return to the batch file.
PUT LOGOUT
;logout on the remote site
LOG OFF
DISCONNECT
;disconnect the phone line at the remote site
QUIT
;end the TRANSIT script and return to CONNECT.BAT
```

Batch Commands

A batch file consists of programs to execute and special batch file commands. The following section describes each batch command.

@

Syntax: `@command`

When a batch file line begins with the @ character, the batch processor does not echo the line on the console regardless of the current state of the echo flag. Many batch files begin with the line:

```
@ECHO OFF
```

The leading @ means the ECHO command is not displayed, and the ECHO command itself turns OFF the echo flag so subsequent batch file lines are not displayed either.

AND

Syntax: `IF [NOT] condition [AND [NOT] condition] Command`

AND is a command that is used in conjunction with the IF batch command to take (or not take) some action based on the state (true or false) of two or more specified conditions. When AND is used, all of the of the specified conditions must be true in order for the action to be taken. AND is used as the operator in the following:

If Condition1 operator [AND] Condition2, then action1 taken.

If condition1 is true and condition2 is true, then action1 executes. If either is false, action1 is not taken. For example, if the CCI11 variable is HOST and if the CCI12 variable is FG, then the phrase `Host foreground` displayes.

```
IF %CCI11% == HOST AND %CCI12% == FG ECHO Host foreground
```

CALL

Syntax: `CALL FILENAME[.BAT]`

The CALL command allows one batch file to execute another batch file. After a CALL'ed batch file is executed, control returns to the original batch file, which then executes the next command in the file. For example, if the SAMPLE.BAT file includes:

```
CALL NEXT.BAT
CALL FINAL.BAT
```

when SAMPLE is executed, the batch file NEXT is called and executed. When NEXT exits, the batch file FINAL is called and executed. When FINAL exits, any remaining commands in SAMPLE are executed.

CALL'ing a second batch file is different than *chaining* to a second batch file. For example, the batch file line:

```
NEXT.BAT
```

chains to the batch file NEXT. A chained batch file does not return control to the original batch file, so that when NEXT is complete or exits, the command line prompt is returned.

CASE

Syntax: `CASE [ON | OFF]`

The CASE command sets the case sensitivity flag for the batch file IF statement. CASE can be turned ON or OFF. Entering CASE with no parameters displays the current value of the case flag. By default, CASE is ON.

Normally, when comparing two strings with the IF command, if the strings are different in any way, including their case, the batch processor thinks the strings are not equal. In the following, string ABC does not equal string abc. When CASE is turned OFF, the batch processor ignores case differences in two strings, returning a "TRUE" condition and displaying the words "Strings Match."

```
CASE OFF
IF "ABC" == "abc" ECHO Strings Match!
```

The CASE flag is useful when comparing %N command line parameters to options in a batch file. With CASE OFF, it is irrelevant if the command line parameter was typed in upper or lower case. Turn CASE OFF and do a single IF string comparison in the BATCH file.

DIREXIST

Syntax: `IF [NOT] DIREXIST dirname command`

DIREXIST is an option to the IF batch processing command. It allows conditional execution of a command based on the existence of a directory. Dirname can be a "full" path specification. The following batch line exam-

ple creates, using the MD utility, a TEMP subdirectory if such a subdirectory does not already exist in the root.

```
IF NOT DIREXIST c:\temp MD c:\temp
```

ECHO

Syntax: `ECHO [ON | OFF | message]`

The ECHO command turns the echo flag to ON or OFF. If ECHO is ON, lines in a batch file are displayed as they are executed. If ECHO is OFF, the batch processor does not display batch file lines as they are executed. ECHO with no parameters displays the current state of the echo flag. The default is ECHO ON.

ECHO also has a message option which displays a message regardless of whether ECHO is ON or OFF.

```
ECHO OFF
ECHO This message is displayed.
```

A blank message can be displayed by following the ECHO command with two spaces or by immediately following the ECHO command with a period:

```
ECHO A blank line separates -
ECHO.
ECHO - these two messages.
```

ECHOERR

Syntax: `ECHOERR [ON|OFF|message]`

This batch command controls whether batch command error messages, or other messages are displayed. The following line causes batch command error messages to be suppressed.

```
ECHOERR OFF
```

EXIT

Syntax: `EXIT`

The EXIT command terminates the execution of a batch file. Normally, execution terminates after the batch processor executes the last command in the batch file. The EXIT command allows the termination of batch processing at any point. For example, to terminate the batch file if the VER environment variable returns an older operating system version number use

```
IF "%VER%" LT "3.05" EXIT
```

This command states that if the VER environment variable is less than 3.05, terminate the batch file.

FOR

Syntax: `FOR %%v IN (set) DO command %%v`

The FOR command allows interative execution of another command. Each entry in the set following IN is substituted into the %% variable used in the command following DO. The %% variable name can be any single alphabetic character.

If the set consists of filenames, wildcard characters (* and ?) can be used to specify a group of files. However, filenames in the set cannot include path names.

The FOR command is also useful from the command line. When using FOR from the command line, use a single percent sign (%) instead of two percent signs (%%) in the FOR variable.

- *Example 1: Copy all files with the .ASM extension and all files with the .OBJ extension to the TEMP subdirectory, one after another.*

  ```
  FOR %%F IN (*.ASM *.OBJ) DO COPY %%F C:\TEMP
  ```

- *Example 2: Compile all .C programs in the current directory, one after another.*

  ```
  FOR %%F IN (*.C) DO CC %%F
  ```

- *Example 3: From the command line, run CHKDSK on drives C:, D:, and E:*

  ```
  C>FOR %D IN (C D E) DO CHKDSK %D:
  ```

GOTO

Syntax: `GOTO label`

The GOTO command turns control over to the line following the GOTO label.

```
GOTO ALABEL
rem this could be one or many lines
rem between GOTO and its label.
:ALABEL
rem these commands are now executed.
```

The first line causes execution of the batch file to start at the line following the label ALABEL. Labels are always preceded by a colon (:) as the first character in the line, and are never displayed during execution of the batch file. Labels can be longer than eight characters, but the first eight characters of a label must be unique in order for the label to be found by the GOTO statement.

A line beginning with a colon (:) can also be used as a comment line.

GOSUB

Syntax: `GOSUB label`

GOSUB transfers control of the batch file to the line following the specified label and returns to the line following the GOSUB command when a RETURN command is encountered. A RETURN command must be specified with GOSUB, so that processing can return after the subroutine identified by the label is executed. The maximum nesting level of the GOSUB command is four levels deep. The following example uses GOSUB to type out two lists before executing the summary routine.

```
@ECHO OFF
REMDisplay the file, intro.txt, then execute a subroutine
to display
REM"all" the lists before calling the batch file,
summary.bat.
Type intro.txt
GOSUB lists
CALL summary
EXIT
:lists
TYPE: list-1.txt
TYPE list-2.txt
RETURN
```

IF

Syntax: `IF [NOT] condition command`

The IF command allows conditional processing of commands. If the IF condition is true, the command is executed. The IF condition is one of the following, and can be combined with a "NOT".

- "string1" relop "string2"
- ERRORLEVEL [==] number
- EXIST filespec
- KEY ["message"] [==] character
- NOTEXIST filespec

The string comparison condition is true if string1 and string2 match according to the relational operator (relop). The allowable operators are:

- == Equals
- EQ (same as ==)
- NE Not equal
- LT Less than
- GT Greater than
- LE Less than or equal
- GE Greater than or equal

When comparing strings, a delimiter character should be placed around the string to delineate the text being compared. For example,

```
IF "string1" == "string2" command
```

is better than:

```
IF string1 == string2 command
```

The IF ERRORLEVEL condition is true if the previous program had an exit code greater than or equal to "number".

The IF EXIST condition is true if the specified file is present on the disk.

The IF KEY condition optionally displays "message" and waits for a keypress. The KEY condition returns true if the keypress is equal to the specified "character".

Test for the false condition by using the NOT option; for example, "NOT condition" returns a true status if the condition is false.

- *Example 1: Use the ERRORLEVEL returned by the OPENS utility to display a message if there are any open files on the system.*

```
OPENS
IF ERRORLEVEL == 1 ECHO There are open files!
```

- *Example 2: Use EXIST to execute a file COPY only if the specified file actually is present on the disk.*

```
IF EXIST THISFILE COPY THISFILE D:\BACKUP
```

- *Example 3: Use a string equality test to type a file only if Carol is the user at this station. If Carol is not the current USERNAME, the type command is ignored.*

```
IF "%USERNAME%" == "CAROL" TYPE OPENUP.TXT
```

- *Example 4: Use DATE to set the date environment variables, then use a string comparison test to execute a backup depending on which quarter of the year we are in:*

```
DATE /E /B
SET QUARTER=Q1
IF "%DOY%" GT "091" SET QUARTER=Q2
IF "%DOY%" GT "182" SET QUARTER=Q3
IF "%DOY%" GT "273" SET QUARTER=Q4
BACKUP %QUARTER%.DAT A:
```

- *Example 5: Use a KEY condition to exit a batch file depending on what the operator says.*

```
IF KEY "Exit batch file (Y/N) ? " == Y EXIT
```

INPUT

Syntax: `INPUT ["message"] envname`

The INPUT command accepts a string from the keyboard (terminated by a carriage return) and assigns it to the environment variable envname. The optional message string can prompt the user for input.

■ *Example: Ask the user to enter the filename to backup to floppy disk. If a filename is entered, back it up. After the backup, clear the variable from the environment.*

```
INPUT "Please enter filename to backup: BKUPFN
IF "%BKUPFN%" NE "" BACKUP %BKUPFN% A:
SET BKUPFN=
```

INPUTC

Syntax: `INPUTC ["message"] envname`

The INPUTC command accepts a single character from the keyboard and assigns it to the environment variable envname. The optional "message" string can prompt the user for input.

■ *Example: ask the user to enter a disk drive to run through CHKDSK.*

```
INPUTC "Please enter a drive letter: " DISKLETTER
CHKDSK %DISKLETTER%:
SET DISKLETTER=
```

INPUTX

Syntax: `INPUTX ["message"] envname`

The INPUTX command accepts a single character from the keyboard and assigns its numerical value to the environment variable envname. The optional "message" string can prompt the user for input.

■ *Example: ask the user to hit F1 to continue*

```
INPUTX "Hit F1 to continue" FKEY
IF %FKEY% == 187 GOTO F1
```

ON

Syntax: `ON [ERRORLEVEL] # [GOTO] label`

The ON statement goes to labels depending on the DOS ERRORLEVEL. The ON statement can be very similar to the IF ERRORLEVEL statement:

```
ON ERRORLEVEL 2 GOTO ERROR_2
IF ERRORLEVEL 1 GOTO ERROR_1
```

However, when used in its abbreviated form, the ON statement can be clearer and easier to write:

```
ON 2 ERROR_2
ON 1 ERROR_1
```

OR

Syntax: `IF [NOT] condition OR [NOT] condition Command`

OR is a command that can be used in conjunction with the IF batch command to take (or not take) some action based on the state (true or false) of two or more specified conditions. When OR is used, one or more of the specified conditions must be true in order for the action to be taken. OR is used as the operator in the following:

■ *Example: If condition1 [OR] condition2 then action1 taken.*

If condition1 is true or condition2 is true or both are true, then action 1 takes place. In the example below, if the CCI11 variable is HOST or if the CCI12 variable is FG, then the phrase `Host foreground` displays.

```
IF %CCI11% == HOST OR %CCI12% == FG ECHO Host foregrour1
```

PAUSE

Syntax: `PAUSE [prompt]`

The PAUSE command temporarily halts execution of a batch file while some external action is taken. If the ECHO flag is ON, any "prompt" after the PAUSE command is displayed and can be used as a prompt. When the PAUSE command executes, the message `Strike a key when ready` is displayed. Pressing CTRL-C terminates the batch file. Any other keystroke resumes execution of the next command in the batch file.

■ *Example: Use PAUSE to ask the user to check to be sure all users are off the system prior to beginning a backup.*

```
PAUSE Press ^C now if anybody else is logged in
BACKUP C:*.DAT A: /S
```

PAUSEERR

Syntax: `PAUSEERR [prompt]`

This command stops batch file execution until the user has pressed any key to continue. The prompt `Strike any key when ready` is displayed. This prompt is displayed regardless of whether ECHO ON has been explicitly set. The following example causes an error message to be displayed when the drive is not ready.

```
PAUSEERR The %1 drive is NOT ready.
```

REM

Syntax: `REM [remark]`

The REM command adds comments (up to 123 characters,) to the batch files to explain what they do. If the ECHO flag is ON, remarks are displayed as the batch file executes. Lines beginning with colons (:) and semi-colons (;) also add comments to a batch file. Comment lines beginning with REM

or semi-colons ECHO whereas comment lines beginning with colons do not.

RETURN

Syntax: RETURN

This batch command is used in conjunction with a GOSUB or SWITCH command. The GOSUB "label" or SWITCH "label" commands cause the batch file execution to continue with the lines following the referenced "label". The RETURN command then causes the processing of the batch file to return to the line following the GOSUB or SWITCH command, and continue execution from that point on. See the GOSUB or SWITCH commands for examples of batch files using RETURN.

SHIFT

Syntax: SHIFT

The SHIFT command causes the command line replacement variables %1 through %9 to be shifted to the left so more than 10 command line parameters can be accessed within a batch file. After a SHIFT, the %1 variable contains the old %2 variable's value, and so on, until %9, which was the next, (and previously inaccessible) command line parameter.

- *Example 1: Invoke the batch file JOE.BAT with 11 command line parameters, A through K. Inside JOE.BAT, use SHIFT to access the tenth and eleventh command line parameters J and K.*

```
C>JOE.BAT A B C D E F G H I J K

REM joe.bat
ECHO %0 is JOE.BAT,  %1 is A,  %9 is I
SHIFT
ECHO %0 is A,  %1 is B,  %9 is J
SHIFT
ECHO %0 is B,  %1 is C, %9 is K
```

- *Example 2: Use SHIFT and an IF string comparison to sequentially process command line arguments until there is a blank argument. This technique allows a batch file to process a variable number of command line arguments.*

```
:LOOP
IF "%1" == "" GOTO DONE
ECHO Now processing "%1"
REM Do processing here
REM Shift next argument into %1
SHIFT
GOTO LOOP
:DONE
ECHO Done
```

SWITCH

Syntax: `SWITCH label1, label2,...labeln`

SWITCH allows the choose between different routines within the "same" batch file, where each routine is indicated by a unique label. It prompts for a keystroke indicating the desired label number (1-9). The maximum number of labels for a SWITCH command is 9. A batch processing example:

```
@ECHO OFF
REM      Display the file intro.txt, then execute a
REM subroutine to display
REM      the "desired" list before calling the batch file
REM summary.bat.
TYPE intro.txt
SWITCH list1, list2, list3
CALL summary
EXIT
:list1
TYPE list-1.txt
RETURN
:list2
TYPE list-2txt
RETURN
:list3
TYPE list-3.txt
RETURN
```

Chapter 4:
Security System

Security System

The CCI Concurrent DOS operating system includes an optional Security System. This Security system is composed of two parts: System Administration and File Security. System Administration controls access to the host computer by requiring users to supply their names and a password before they can use the system. This is called **Password** security. Or, it may be desired to have users enter their names without requiring them to enter a password. This is called **Name Service.** System Administration also allows designation of which stations and/or ports are password-controlled, what their boot-time environments are and what level of system access logging is done. It can even provide a very high level of access security for remote workstations used in conjunction with the LOOKOUT program, described more fully in the Remote Access Chapter.

The File Security portion of the Security system controls how individual files and directories may or may not be accessed by system users, including Read, Write, Execute and Delete. Every file and directory has those permissions set for the Owner (the person who created or last modified the file), the Group (the group that the owner belongs to) and the World (everyone on the system). Users can each be assigned to one or more primary and secondary groups, up to a maximum of 255 of each.

System Administration can be used with File Security inactive. However, the use of File Security requires System Administration.

The use of security is recommended, if only for name service, since many CCI Concurrent DOS utilities (such as eMAIL and WHO) will take advantage of user names if they are available.

Enabling Security

When installing the CCI Concurrent DOS system, a SECURITY subdirectory is created in the CDOS subdirectory and some important files are copied into it. Security is not enabled automatically during installation; activating it simply requires the renaming of a single file.

In the SECURITY subdirectory, there are the CONTROL.OLD and CONTROL.DOC files, the SECURITY.EXE program, and a sample LOGIN.BAT file. The CONTROL files are security configuration files. CONTROL.DOC is a fully commented file with an explanation of each available option. To activate the Security system rename CONTROL.OLD to CONTROL. SECURITY.EXE is used to assign each user a name and password, boot-time and login options, and group options necessary for file security. LOGIN.BAT is analogous to a START.BAT file.

When security is active, the system executes NAME.BAT files when each person (NAME) logs in. If a NAME.BAT file does not exist, the LOGIN.BAT

file found in the SECURITY subdirectory executes. Create optional NAME.BAT and NAMEy.BAT files to set the environment (for example, printer, path, program, etc.) for each user by name or a generic LOGIN.BAT file to be executed whenever a user logs in.

When the security system use begins, the operating system creates a UTMP.DAT file that contains a list of currently logged in users. This file, created in the root directory, is used by other Concurrent Controls utilities, including WHO, LOGIN, and LOGOUT. When the security system is activated, it also creates a CCIUSERS.SYS file containing all user information, a HISTORY.DAT file that logs the time each station is powered on and all logins and logouts, and a SYSADMIN.DAT file that logs all system accesses.

The security system is enabled when a CONTROL file (no extension) is found in the SECURITY subdirectory at boot. Therefore, renaming CONTROL.OLD to CONTROL and rebooting activates the basic system administration options of the security system. Additional options, such as FILE-SECURITY, can be added to CONTROL if desired. However, before doing this, it is necessary to first enter system users, passwords (if desired), and login defaults using the SECURITY program.

User Parameters and Password Maintenance

The SECURITY program adds authorized usernames to the system and defines a variety of conditions and parameters that affect the user's environment, default settings, and access to the system. These include:

- The default printer
- Drive and path
- Whether START and NAME files execute
- Which stations a user is allowed to login on
- Original passwords
- The group and default privileges a user is assigned to (only with file security active)

All of this information is stored in CCIUSERS.SYS, an encrypted file. To use SECURITY, change directories to C:\CDOS\SECURITY and enter SECURITY at the command prompt. The following screen is displayed:

```
                    SECURITY.EXE
          CCI Security Files Maintenance Utility
     Copyright (C) 1992-1994 by Concurrent Controls, Inc.
                (CCI Release:  02/04/94)

This utility maintains user information for the CCI Concurrent DOS system.

     Entries FETCHED:  CCIUSERS.INF is 002, CCIUSERS.SYS is 002,
                       CCIGROUP.SYS is 004, CCIORGAN.SYS is 000.

          (G)  GROUPS.          | Setup Groups, then        |
          (U)  USERS.           | setup Users, and then     |
          (O)  ORGANIZATION.    | setup the Organization.   |
           .   (A)  ADD.        |                           |
           .   (M)  MODIFY.     | Select G, U or O and an   |
           .   (D)  DELETE.     | Action A, M, D, V or P.   |
           .   (V)  VIEW.       | Repeat, as desired.       |
           .   (P)  PRINT.      |                           |
          (S)  SAVE CCI Security Files, then EXIT.
          (X)  EXIT Immediately.
          (H)  Display the SECURITY HELP Screens.
          (?)  Display this MENU Listing.

                    OPTIONS : ACTIONS
          SELECTION: (?) -
```

Select the (U)sers Maintenance option, then select the ADD option to add a name and password for each user. When finished, use the PRINT option to print out a hard copy to be stored in a secure place. Then use the SAVE option to save the names and passwords entered in the CCIUSERS.SYS file. The CCIUSERS.SYS file includes a variety of information for each user entered. The following is a list of those fields, their defaults, and field types.

Field	Default	Explanation
Username:	REQUIRED	Only required if File Security is needed. Unique name, maximum of eight alphanumeric characters.
Primary:	FF-NONE	Enter hex ID for "home" primary group name. If not using File Security, just press Enter.
Secondary:	FF-NONE	Only required if File Security is needed. Enter hex ID for "home" secondary group name.
Password:	NONE	Maximum of eight alphanumeric characters. If name service only for this user, just press Enter.
Def. Access:	FULL	Required only if File Security is needed. Enter the default file access permissions for Owner, Group & World,

		which are automatically applied to a file or directory when created. Full access, RWEDRWEDRWED, is the default.
Drive:Path:	BOOTROOT	Enter the desired drive:path to go after LOGIN. Specify BOOTROOT to use boot-time drive:path, or DEFAULT to maintain last drive:path.
Printer:	000	Enter the desired printer number, to use after LOGIN. Specify LOCAL to use terminal printer number, or DEFAULT to maintain last assigned printer.
Terminals:	ALL	Enter list of workstation numbers that this user is restricted to log onto, separated by commas, where Host=0. Example: 0,1,20.
Command:	NONE	Enter a command or batch file to execute, with pathing, if necessary, to automatically execute on LOGIN. When the specified command or batch file is done, the session is automatically logged out and returned to a log in prompt. Auto LOGOUT is done.
STARTxxy:	YES	Run STARTxxy.BAT files on LOGIN? Enter Y/N.
NAMEy:	YES	Run NAMEy.BAT files on LOGIN? Enter Y/N.
Reset:	YES	Reset the environment? Enter Y/N.
Comment:	NONE	Just a comment; field not used elsewhere.

Enter a period (.) to reset a field to its default values, except for username. When the user information entry is complete, (S)ave the entries and e(X)it the SECURITY program.

The following is an explanation of each field and its use.

The **Username** field is the name the user enters at a `Username` prompt. A username can be up to eight characters long. Each user must have a unique username.

The **Primary** field is the "home" primary group name; the **Secondary** field is the "home" secondary group name. This parameter is only used with file security; if not using file security, hit Return at this field.

The **Def. Access** field is the default file access permissions for Owner, Group, and World that are automatically applied to any file or directory created by this user. Full access, RWEDRWEDRWED, is the default. See the section on File Security later in this chapter for additional details.

The **Password** field is the password the user enters at a "Password" prompt during LOGIN. If name service without passwords is to be used, press Enter to skip the password field. If passwords are being used and the password field is skipped, the security shell thinks this user doesn't need a password on LOGIN and displays `NONE REQUIRED` at the password prompt. A password can be up to eight characters long. Passwords should be something that can be easily remembered by the user, but not something which can be easily guessed. In high-security installations, passwords are changed on a regular basis and consist of random characters or nonsense words that can not be found in a dictionary.

The **Drive:Path** field is the user's default path. When the LOGIN process finishes, the user is placed in this drive and subdirectory. Options available are BOOTROOT, DEFAULT and explicit drive:path assignments. The default option is BOOTROOT. The BOOTROOT option results in the boot-time value of the drive, path and SUBST command being used upon login. The DEFAULT option results in the value of the drive, path and SUBST command from the previous user being used upon login. An explicit drive:path can also be entered and will be used if valid. Alternatively, use a CHDIR command in the LOGIN.BAT or the user's NAMEy.BAT file to put the user in a specific drive and subdirectory if the location to put them in changes frequently.

The **Printer** field is the user's default printer. Options available for this field are LOCAL, DEFAULT and explicit printer number. LOCAL is the default. The LOCAL option results in the workstation printer number being assigned upon login. Use of the DEFAULT option results in the printer assignment from the previous user being assigned upon login. An explicit printer number can be entered for this field. However, if the number entered is not a valid printer, upon login, the user is assigned to printer 00. Alternatively, use the PRINTER command in a LOGIN.BAT or NAMEy.BAT file, if the user's default printer number is apt to change frequently.

The **Terminals** field restricts the user to access one or more specified workstations. Enter the station numbers that the user can log into, separating each number with a space. The host console is workstation 0, the first terminal is workstation 1, etc. This is very useful if a menu system is being used to control access.

The **Command** field is the name of a program to execute on all sessions when the user logs in. When the program is terminated, the user is automatically logged out.

The **Comment** field is used by the system administrator to attach a comment to this entry. This field is not used by any other utility.

The **STARTxxy** field determines whether the STARTxxy.BAT files are run when this user logs in. This field is usually set to Yes.

The **NAMEy** field determines whether a NAMEy.BAT file should be executed for this user. This field is usually set to Yes.

NOTE: A START000.BAT must exist for NAME.BAT to run.

The **Reset** field is a Yes or No field determining whether the command shell should be reset to the boot-time condition when this user logs in. This field is usually set to Yes. See the &COMMANDS in the CCI Concurrent DOS Utilities and Troubleshooting Guide, for further information on clearing a user's environment.

After user information has been entered, the CONTROL file is then modified and renamed, if necessary. The CONTROL file is an ASCII text file that contains only the options needed and has minimal content (as in the CONTROL.OLD file) to ensure fastest processing at system boot time. The next section describes the available CONTROL file options.

CONTROL Configuration File

The contents of the CONTROL file in the SECURITY subdirectory determine which features of the Security System are activated. The CONTROL file can be edited using the CCI EDITOR or other ASCII file text editor, or a word processor that supports a text mode. When editing, start with the CONTROL.OLD file supplied with the operating system and modify to include the needed options.

Entries in the CONTROL file fall into one of three general areas: Boot-time, Login, or Reporting. Boot-time options take affect at boot-time and range from how the system boots (single user or multiuser) to the level of security invoked (password, file security). The Login options affect the user's login process and the environment at login time. Reporting options affect what and when is written to one of two activity logs: HISTORY.DAT and SYSADMIN.DAT. Two of the reporting options, LOG-BAD-NAMES and LOG-BAD-PASSWORDS, have an additional option of terminating a dial-in connection.

The possible entries in CONTROL and their general categories are:

SINGLE or MULTI	Boot-time option
DO-UPON-BOOT	Boot-time option
FILE SECURITY	Boot-time option

*AUTO-LOGIN or PASSWORD	Boot-time option
*HISTORY	Reporting option
*START-FIRST	Login option
*BLIND-LOGIN	Login option
*DOUBLE-BLIND-LOGIN	Login option
*CLEAR BUFFERS	Login option
System Administration Options	
*SYS-FILE	Reporting option
*SYS-PRINTER	Reporting option
*LOG-BAD-NAMES	Reporting and action option
*LOG-BAD-PASSWORDS	Reporting and action option
*LOG-DISCONNECTS	Reporting option
*LOG-LOGINS	Reporting option
*LOG-LOGOUTS	Reporting option

The lines marked with an asterisk (*) have two additional modifiers: INCLUDE and EXCLUDE.

INCLUDE and EXCLUDE

The presence of INCLUDE and/or EXCLUDE modifies the workstation or user numbers to which the previous option in the CONTROL file apply. If INCLUDE or EXCLUDE is not present, the previous option applies globally to all users on all workstations. The INCLUDE or EXCLUDE command is followed by a list of workstation numbers (0, 1, 2, etc.) or user numbers (u1, u2, u3, etc.) to be included or excluded from the previous option. Since user numbers tend to be more dynamic than station numbers, the latter is recommended. The word ALL can follow INCLUDE or EXCLUDE to mean all users on all workstations.

INCLUDE and EXCLUDE can appear together after a single CONTROL file option. For example,if the PASSWORD option should apply only to the main console and the first station, use the following:

```
PASSWORD
EXCLUDE ALL
INCLUDE 1
```

Text after a semicolon (;) is ignored and can be used to include comments in the CONTROL file. The following is a detailed description of the other CONTROL file options.

Boot-time options

SINGLE

The SINGLE option enables single-user operation when the system is first started. Only the first session on the host console is usable. This option allows a system administrator to perform backups or changes in other files, knowing no other user is logged into the system. When the host console issues a LOGOUT, normal multi-user operation resumes. To boot in normal multiuser mode, use the MULTI option instead.

MULTI

The MULTI option enables normal multiuser operation. After the system is booted, all workstations are usable. To boot the system in a single user mode, use the SINGLE option instead. MULTI is the default.

FILE SECURITY

The FILE SECURITY option enables the file security system. This line must be included to control user access privileges to files and directories. The File Security portion of the Security system is covered in detail in the second section of this chapter.

PASSWORD

Enables name service and optional password protection. Can be followed by INCLUDE or EXCLUDE to specify which consoles use passwords. After the system is booted, if the CCI USERS.SYS PASSWD file exists, the selected consoles automatically perform a LOGIN command to ask for a username and password. To select name service only, see AUTO-LOGIN. The PASSWORD option takes precedence over AUTO-LOGIN if both options are specified for a particular console.

Login Options

AUTO-LOGIN

Enables name service without password protection. Can be followed by INCLUDE or EXCLUDE to specify which consoles use name service. After the system is booted, the selected consoles automatically perform a LOGIN command to ask for a username. If AUTO-LOGIN is not in the CONTROL file, all workstations display the command line prompt after the system is booted. Use the PASSWORD option to have names and passwords on LOGIN.

DO-UPON-BOOT Un

This causes the CCIBOOT.BAT file to run upon system boot. The CCIBOOT.BAT batch file must be in the SECURITY subdirectory, and runs on only one user (Un, where n is the user number). This allows a series of commands, such as print spooler start or file server initialization, to be executed before normal login procedures.

START-FIRST

This option selects whether the STARTxxy.BAT files are run before the LOGIN prompt. When the system boots, the default is to run the STARTxxy.BAT files first and run the LOGIN.BAT or NAMEy.BAT files after a LOGIN. Use this option with EXCLUDE ALL to defer execution of the STARTxxy.BAT files until after a LOGIN.

BLIND-LOGIN

This option eliminates the normal LOGIN prompts of

```
Please press RETURN TWICE for login
To LOGIN, please enter your Name and Password
```

prompts and replaces both the `Username` and `Password` prompts to a simple ? character on consoles that require a login name and password. If an invalid username or password is entered, the `INVALID` response is also suppressed.

This option can be used on modem channels to restrict the amount of information displayed on a LOGIN. If someone is trying to break into the system and they see a `Username` or `Password` prompt, it gives them an idea of what to enter. The BLIND-LOGIN option requires someone dialing in to know the sequence of keystrokes required to log into a CCI Concurrent DOS system: two carriage returns, a username, a carriage return, a password, and a final carriage return.

DOUBLE-BLIND-LOGIN

This option eliminates the default LOGIN prompts entirely. If an invalid username or password is entered, the `INVALID` response is also suppressed. See BLIND- LOGIN for more information.

CLEAR-BUFFERS or NO-CLEAR-BUFFERS

These determine if the command shell clears the console's command line history buffers after a LOGIN. The consoles must also have the PASSWORD option enabled. CLEAR-BUFFERS is the default.

Reporting Options

HISTORY

If the HISTORY option is present, the system records when the system is started and when users login and logout in a file called HISTORY.DAT. The HISTORY option can be followed by NEW or APPEND. The HISTORY NEW option deletes and recreates the HISTORY.DAT file each time the system is started. The HISTORY APPEND option keeps a cumulative record in HISTORY.DAT until the file is erased by the system administrator. The default is HISTORY APPEND.

CAUTION: *If the APPEND option is used, the system administrator should periodically monitor the size of the HISTORY.DAT file. If it becomes too large it should be copied elsewhere or erased. The format of the HISTORY.DAT file follows.*

```
"CAROL   ","00","LOG-OUT   ","05/22/93 08:02:52","HOST"
"CAROL   ","00","LOG-IN    ","05/22/93 08:02:58","HOST"
"NORM    ","12","LOG-IN    ","05/22/93 08:03:40","A8"
"GREG    ","01","LOG-OUT   ","05/22/93 08:04:31","P1"
"GREG    ","01","LOG-IN    ","05/22/93 08:04:33","P1"
"MARIA   ","08","LOG-IN    ","05/22/93 08:06:37","A4"
"********";"02","ACTIVATED ","05/22/93 08:06:46","P2"
"ENRIQUE ","02","LOG-IN    ","05/22/93 08:06:56","P2"
"********","04","ACTIVATED ","05/22/93 10:06:08","P4"
```

The first field is the user's name, the second field is the workstation number they are using, the third field is the activity being logged, the fourth field is the date and time the activity took place, and the last field is the Concurrent Controls port name (see the CCI11 variable in the &SETCCI command in the *CCI Concurrent DOS Utilities and Troubleshooting Guide.*).

CAUTION: *Do* **not** *edit the HISTORY.DAT file in the SECURITY subdirectory. The HISTORY.DAT file must be available for the system to record user logins and logouts. Also, editing the HISTORY.DAT file with a word processor may leave the file in an unknown state, possibly inserting invalid control characters. Copy HISTORY.DAT to a different filename or subdirectory before doing any type of processing.*

System Administration Log Options

The remaining CONTROL file reporting options are used by system administrators to collect more information about login activity. The LOG- options specify what information is saved. The SYS- options specify where the information is saved. The LOG- options can be followed by one or two optional parameters:

SYS-FILE — Save the LOG- option activity to a SYSADMIN.DAT file. This is the default.

SYS-PRINTER — Print the LOG- option activity on a printer.

LOG-DISCONNECTS

Record disconnect times. If a user is calling in on a modem channel, and the channel disconnects, the disconnect is recorded.

LOG-BAD-NAMES

Record any attempts to login with an illegal user name. This option can be followed by optional AFTER # and DISCONNECT parameters. The AFTER # parameter specifies how many attempts are allowed before the illegal attempt is recorded. Three tries is the default. The optional DISCONNECT allows the system to disconnect the modem after the number of attempts have expired. If the person trying to illegally login is calling in on a modem line, this hangs-up the modem. If the HISTORY.DAT or SYSADMIN.DAT file contains BAD-NAME login activity, it could mean someone is trying to break into the system.

LOG-BAD-PASSWORDS

Record any attempts to login with an illegal password. This option also can be followed by the AFTER # and DISCONNECT parameters used in the LOG- BAD-NAMES option above.

LOG-LOGINS

Record all login attempts to the SYSADMIN.DAT file or to the SYS-PRINTER printer.

LOG-LOGOUTS

Record all logouts to the SYSADMIN.DAT file or to the SYS-PRINTER printer. LOG-LOGOUTS can be followed by an optional DISCONNECT parameter to hang-up a modem if the LOGOUT occurs on a dial-in phone line.

SYS-FILE

When used after one of the above LOG- options, SYS-FILE directs the operating system to record the LOG- activity in the SYSADMIN.DAT file. When used alone, with the NEW or APPEND options, SYS-FILE directs the operating system to either create a new SYSADMIN.DAT file when the system is started, or allow the LOG- options to append any information to an existing SYSADMIN.DAT file. SYS-FILE APPEND is the default.

SYS-PRINTER #

When used after one of the above LOG- options, SYS-PRINTER directs the operating system to record the LOG- activity on a printer. When used alone, with a printer number, SYS-PRINTER specifies the printer where LOG- activity should be printed. SYS-PRINTER 0 is the default.

Using the File Security System

The Concurrent Controls file security system controls how files may or may not be accessed by system users. It can be effectively used to limit certain users or groups of users to the use of only those files necessary for their jobs.

Permissions and Levels

The different ways (or permissions) that files can be used are Read, Write, Execute and Delete. Every file and directory has those permissions set for the Owner (the person who created or last modified the file), the Group (the group that the owner belongs to) and the World (everyone on the system). Those permissions are set at creation or modified thereafter with the XATTRIB utility and are represented as follows, with four permissions possible at each level: RWED RWED RWED

Every system also has a SYSOP (SYStem OPerator) to configure file security and have access to all files and directories on it.

Groups

A group is one or more users, and has an alphanumeric name (maximum of eight characters) and an ID created in the SECURITY program. The file security system provides for both primary and secondary groups, and allows a user to be assigned to a pair — one primary and one secondary at any point in time. The user can belong to as many primary and secondary groups as the system supports and moves among them as necessary. The system supports as many as 255 primary and 255 secondary groups.

Groups are defined and members added using portions of the SECURITY program discussed earlier. This is generally done by the SYSOP, who should "own" SECURITY.EXE. That file should also be given access permissions of "RWED ——— ———" so that no one but the OWNER can access it.

The Security program maintains user and password information, as well as group and organization detail. An example SECURITY display follows.

```
                    SECURITY.EXE
          CCI Security Files Maintenance Utility
     Copyright (C) 1992-1994 by Concurrent Controls, Inc.
                (CCI Release:  02/04/94)

This utility maintains user information for the CCI Concurrent DOS system.

     Entries FETCHED:  CCIUSERS.INF is 002, CCIUSERS.SYS is 002,
                       CCIGROUP.SYS is 004, CCIORGAN.SYS is 000.

          (G)  GROUPS.          | Setup Groups, then       |
          (U)  USERS.           | setup Users, and then    |
          (O)  ORGANIZATION.    | setup the Organization.  |
           .   (A)  ADD.        |                          |
           .   (M)  MODIFY.     | Select G, U or O and an  |
           .   (D)  DELETE.     | Action A, M, D, V or P.  |
           .   (V)  VIEW.       | Repeat, as desired.      |
           .   (P)  PRINT.      |                          |
          (S)  SAVE CCI Security Files, then EXIT.
          (X)  EXIT Immediately.
          (H)  Display the SECURITY HELP Screens.
          (?)  Display this MENU Listing.

                    OPTIONS : ACTIONS
          SELECTION: (?) -
```

Groups must be selected before users can be assigned to them, either in SECURITY or through the (O)rganization option. The (G)roup Maintenance option (A)dd provides the following screen:

```
ADD Primary and Secondary Groups:
Press CTRL_X, at any field, to STOP additions.
  1.  P or S:  P        Primary:    01-ADMIN     'Enter' to ADD.     ADDED.
  2.  P or S:  P        Primary:    02-ENG       'Enter' to ADD.     ADDED.
  3.  P or S:  P        Primary:    03-MKTG      'Enter' to ADD.     ADDED.
  4.  P or S:  P        Primary:    04-QA        'Enter' to ADD.     ADDED.
  5.  P or S:  P        Primary:    05-SUPPORT   'Enter' to ADD.     ADDED.
  6.  P or S:  S        Secondary:  01-GENERAL   'Enter' to ADD.     ADDED.
  7.  P or S:  S        Secondary:  02-MANAGER   'Enter' to ADD.     ADDED.
  8.  P or S:  S        Secondary:  03-MISC      'Enter' to ADD.     ADDED.
  9.  P or S:
```

Only the type — (P)rimary or (S)econdary — and the name of the Group being added must be entered here. The ID number, which is expressed in hex, is assigned by the system, but is needed for entry into the (U)ser Maintenance option of the SECURITY program. After the (U)sers Maintenance file has been created or modified (according to the instructions at the beginning of this chapter), each user's file should reflect a PRIMARY and SECONDARY group ID that will be their default groups at time of boot.

The last step necessary is to add users to additional groups, if that is required, using the (O)rganization option of SECURITY. Selecting that option

and choosing the (A)dd option from the Main Menu displays the existing primary and secondary groups and the members of each:

```
ID  Primary      ID  Primary      ID  Primary      ID  Primary      ID  Primary
-- ---------     -- ---------     -- ---------     -- ---------     -- ---------
01-ADMIN         02-ENG           03-MKTG          FF-NONE          04-QA
05-SUPPORT       00-SYSTEM

ID Secondary     ID Secondary     ID Secondary     ID Secondary     ID Secondary
-- ---------     -- ---------     -- ---------     -- ---------     -- ---------
01-GENERAL       02-MANAGER       03-MISC          FF-NONE          00-SYSADMIN


ID Username      ID Username      ID Username      ID Username      ID Username
-- ---------     -- ---------     -- ---------     -- ---------     -- ---------
02-ARKADY        03-BUDDY         04-CAROL         0D-CASSIE        01-CCINET
05-DAWN          06-ENRIQUE       07-GREG          08-JON           09-KAREN
0A-KEN           0B-NORM          0C-RAY           00-SYSOP

ADD Authorized Users for selected Primary/Secondary Group:

Press CTRL_X, at any field, to STOP additions.

ID-Primary:      02-ENG
ID-Secondary:    02-MANAGER

  1. ID-Username:       0C-RAY          'Enter' to ADD.     ADDED.
  2. ID-Username:
```

Users are always added to a pair (one primary, one secondary). The hexidecimal ID (displayed above) is entered at the ID-Primary and ID-Secondary prompts, and also at the ID-Username prompt. When done entering users for that pair, depressing the return key at the prompt displays again the `ID-Primary:` prompt.

After any additions or modifications in any of the maintenance options on the main menu, the (S)ave option must be selected before exiting for the information to be written to disk.

The process of adding user information, setting up groups and adding members to groups creates and/or updates four security system files, namely: CCIUSERS.SYS, CCIGROUP.SYS, CCIORGAN.SYS and CCIUSERS.INF. These files are placed in the \CDOS\SECURITY subdirectory.

- The CCIUSERS.SYS is an encrypted file that associates user ID numbers with passwords, and user initialization data. This file is checked for password verification, and initialization data upon login. It maintains the fields discussed earlier in this chapter.
- The CCIUSERS.INF is an ASCII text file that maintains user account information for each user authorized to use the system.

This file may be read by all users, but should only be modified by the system administrator (SYSOP) using this SECURITY.EXE program.

- The CCIGROUP.SYS is a non-encrypted binary file that associates valid primary group ID numbers and names, and secondary group ID numbers and names.
- The CCIORGAN.SYS is an encrypted file that maintains the organization of users authorized to access specified primary/secondary groups.

All of these files should be set to RWEDR—R— to prevent unauthorized modifications.

User Interface to File Security

The user's file access interface to the system is through the XATTRIB.EXE utility, located in the \CDOS subdirectory. The XATTRIB command displays or changes access permissions of existing files and directories, sets the user's default access permissions that are automatically applied to files and directories the user creates, changes the ownership of a file or directory, and temporarily switches a user's primary/secondary group access to another authorized primary/secondary group.

Entering XATTRIB/H with File Security enabled displays the following screen:

```
XATTRIB R2.00    DISPLAY OR CHANGE FILE ACCESS PERMISSIONS AND OWNERSHIP
Copyright (c) 1991,1992 Digital Research Inc.  All rights reserved.
Portions Copyright (c) 1992 Concurrent Controls, Inc.   CCI Release: 10/01/92

XATTRIB [/Help] [@][d:][path]filename[.ext] [/switches]

The file specification is the drive, path and name of file(s) for which access
permissions and ownership are to be changed (wildcard filenames are allowed).
Multiple files may be specified on the command line.
Use '@' to specify that the filename that follows is the name of a file list.

  /A:NAME        Change file owner to login NAME of the new owner.
  /B             Brief - Display file names only.
  /D             Operate on directories, not files.
  /F             Display or change default file create permissions.

  /OGW:RWED      O-Owner, G-Group and W-World (Left of ':').
                 R-Read, W-Write, E-Execute and D-Delete (Right of ':').
  /P             Prompt for operating on each file, or paginate if display only.
  /S             Search for files in sub-directories.
  /U:NAME        Only operate on files with owner or primary group NAME.
  /X:PRI:SEC     Switch current primary/secondary group access for current user.
```

XATTRIB /A:NAME

The owner of a file or directory is generally the user that created it, though ownership can be changed. A file or directory can only have one owner at a one time. To change ownership of a file or directory, the current owner or SYSOP performs an XATTRIB /A:NAME option.

XATTRIB /X:PRI:SEC

Each user normally belongs to a primary group, and, optionally, a secondary group, that are established by the SYSOP using SECURITY.EXE. This primary and secondary group "pair" is the user's "home" group, established when the user does a LOGIN. The user may switch to different primary/secondary groups provided the SYSOP has "authorized" access to the desired group pair. A user performs an XATTRIB /X:PRI:SEC option to switch to another group pair.

XATTRIB /OGW:RWED

A user changes access permissions of existing files and directories using the XATTRIB /OGW:RWED option. The /class:permission combination is repeated for each different permission the user wishes to establish. The class type may be one or more of the letters O, G and W, for owner, group and world, respectively. The permission type may be none or one or more of the letters R, W, E and D, for read, write, execute and delete.

The existing settings remain in place for any class not specified in the class field. Where a permission is not specified, on the other hand, that permission is actively denied. Where an access permission is not granted, a dash (-) appears in the appropriate location.

For directories, the /D option must also be specified, and further, for each class of user, either total permission (grant), allowing read, write and delete operations (execute has no meaning with directories), or none at all (deny), must be specified.

XATTRIB /F /class:permissions

Default access permissions are established for each user when the user account is first set up by the SYSOP using the SECURITY.EXE command. The XATTRIB /F option lets the user display the user's default access permissions. The XATTRIB /F /class:permissions option changes the default access permissions to those specified with the form /OGW:RWED. Changes to the defaults apply to the current working session only. To make such changes permanent, either put XATTRIB /F settings in a batch file (ie. NAME.BAT), or have the SYSOP make the change in the user account setup with the SECURITY.EXE command.

XATTRIB /D

To display or change ownership of directories ONLY, the XATTRIB /D option is specified in addition to any other options desired. Changing ownership of a directory does not change ownership of files included

within the specified directory, which protects file ownership when more than one user owns files in the specified directory. File ownership must be explicitly changed.

```
XATTRIB /U:NAME
```

To display or operate only on files or directories owned by user NAME or primary group NAME, the XATTRIB /U:NAME option is performed specifying the appropriate NAME. If the user has access permission, the operation is performed. For directories, the /D option must also be specified.

When file security is enabled, the file handling utilities ATTRIB, BACKUP, MOVE, TOUCH, TREE, XCOPY, XDEL and XDIR have an additional switch /U:NAME that limits their use to those files owned by the user or group specified with NAME.

Using the Security System

This section offers some examples of implementing varying levels of options to achieve specific system access and use goals.

■ *Example 1: Enable name service only*

Set up the Security System for name service so users have to login by name. No passwords are required. The CONTROL file contains:

```
MULTI
PASSWORD
INCLUDE ALL
```

The startup sequence is as follows:

1. When the system is booted a UTMP.DAT file is created and the STARTxxy.BAT batch files are run. The system first tries to run a STARTxxy.BAT file for each session (y) on each workstation (xx). If a STARTxxy.BAT file does not exist for a particular session, the system runs the START000.BAT file on that session.

2. The PASSWORD option tells the system to perform a LOGIN command on ALL workstations. LOGIN displays a `Username` prompt and waits for the user to login. PASSWORD requires the entry of user information in SECURITY. However, passwords are not required.

3. When a user logs into a workstation, the UTMP.DAT file is updated and the command shell runs the NAMEy.BAT file for each session (y) on the workstation. If a NAMEy.BAT file does not exist for a particular session, the command shell runs NAME.BAT on that session. If there is no NAME.BAT file, the command shell runs LOGIN.BAT.

4. When the user logs out, the user's UTMP.DAT entry is cleared.

   The SINGLE option tells the system to start up with only the first session on the main console active. Someone on the main console can

login and perform some function which requires no one else is on the system. When that user logs out, normal multiuser operation resume.

- *Example 2: Enable passwords on a dialin port only.*

This example shows how to use both EXCLUDE and INCLUDE after an option to enable passwords only on COM1 (workstation number 1). This provides access security on dial-in modem lines on the COM1 port without impacting the rest of the system.

```
MULTI
PASSWORD
EXCLUDE ALL
INCLUDE 1
```

The "1" represents the station number of the dialin port.

- *Example 3: Set up a secure system with logging of system login and access activity; use CCIBOOT to start fileserver and other one-time functions at boot.*

```
MULTI
START-FIRST
EXCLUDE ALL
DO-UPON-BOOT U4
PASSWORD
HISTORY APPEND
;system administration options
SYS-FILE APPEND
LOG-LOGINS
LOG-LOGOUTS DISCONNECT
LOG-BAD-NAMES AFTER 5 DISCONNECT
LOG-BAD-PASSWORDS AFTER 5 DISCONNECT
LOG-DISCONNECTS
```

DO-UPON-BOOT tells CCI Concurrent DOS to execute the CCIBOOT.BAT file on user number 4 (the 4th session on the host console). The system administration options record all login and logout activity, including attempts to login with an invalid name or password. In addition, modem channels are automatically disconnected after a logout or after five invalid login attempts.

The HISTORY and SYS-FILE options use the APPEND parameter to maintain a cumulative record of user activity in HISTORY.DAT and SYSADMIN.DAT. Periodically, these files should be erased or copied to another filename or subdirectory so they do not grow too large.

Example 4: Limit the user to a single menu of choices

In a secure environment, names and passwords are required, and users presented with a fixed menu as soon as they login and automatically logged out when they exit the menu.

The following example uses the Menuing system to create that menu. First, these lines are added to the user's NAME.BAT.

```
REM domenu.bat
CD \CDOS
&SETCCI
RUNMENU MENU.DAT
CD \
REM end of domenu.bat
```

This batch file should be located in a directory somewhere in the PATH. The SECURITY "Command:" option should specify the name of this batch file. Set the Security "STARTxxy" and "NAMEy" options to "N". The PASSWORD control file should be followed by EXCLUDE U2 U3 U4 if this batch file is on the main console (unless the menu in all four windows is wanted.)

This page intentionally left blank.

Chapter 5: Remote Access, Support and Communication

Remote Access, Support and Communication

Remote Access

Every CCI Concurrent DOS system supports remote workstations, which are workstations connected through modems. Unlike a network or other DOS environments, no additional software is needed for those remote workstations. A remote workstation can be connected; the dial-in port secured from unauthorized use; files transferred back and forth; dial-in and transfer operations automated; and diagnostics, maintenance, and even application training performed remotely with the functions covered in this chapter. The remote access and support subsystem provides power, flexibility and control not available in any other microcomputer operating system. Quick step-by-step setup instructions follow this introduction, but the entire section should be read for a full understanding of the capabilities included.

The multitasking features of CCI Concurrent DOS mean that communications with other computers can take place while other programs are executing. CCI Concurrent DOS supports TRANSIT (the supplied communications program), several third-party communications programs that are designed for multitasking, multiuser environments, and most DOS communications programs. The highest system performance is delivered using the first two options.

The TRANSIT program and script language included with CCI Concurrent DOS can perform automatic dialing and transfer operations, but do not provide interactive communications. If a communications program with more features and a more sophisticated script language is needed, choose a communications program that is "multiuser aware" for best performance. Included in that category are MLINK from Corporate Microsystems (603-448-5193) and BLAST from Communications Research Group (504-923-0888).

Modem, Remote Workstations and Communications

CCI Concurrent DOS includes full support and software necessary for remote access. Remote workstations can be terminals or computers using a terminal emulator. The type of station required locally (that is, a PC terminal for DOS software, an ASCII terminal for well-behaved programs) will also be necessary at the remote location. Multiple stations in a single remote location are supported using a multiplexer.

A remote workstation requires a modem at the main system and a modem at the remote site. For best performance, the use of at least 9600 baud error checking modems is recommended. Both internal and external modems are supported. External modems are easier to support and troubleshoot be-

cause they have lights to indicate line conditions and can be reset independent of the computer. Multitech and Practical Peripherals modems are recommended for remote access.

A remote workstation dials the phone to connect to the main system. If a computer is being used as the workstation, then the CCI TERM emulation program that comes with CCI Concurrent DOS includes the necessary autodial function. If the remote workstation is a PC terminal, there are several alternatives to choose from:

- Most PC terminals cannot issue an ASCII sequence while in PC terminal mode. This requires setting the terminal into native mode, dialing the phone, and then putting the terminal back into PC terminal mode for the connection. This is not a recommended procedure.
- Some terminals (Link MC5, Relisys) have special key sequences that initiate the dialing of the modem.
- Some modems contain an autodial function.
- Some modems automatically dial the phone whenever the DTR line is raised. The DTR line raises whenever the terminal is turned on.

Next, consider the security of the channel that the remote site is dialing into. Anytime a serial port is available for dial-in access, it should be protected from unauthorized use with physical or system security. The most basic protection method is turning the modem on at the host computer only when it is going to be accessed, and turning it immediately off when done. This is not practical in most situations and a better, flexible, and comprehensive method is provided as part of the CCI Concurrent DOS Security System.

By activating security and requiring a login name and password from anyone dialing into that port, unauthorized access is easily prevented. The BLIND or DOUBLE-BLIND options can limit the display on that dial-in channel to a question mark, or just a blinking cursor. If it is not convenient or necessary to have a login and password at the other stations on the system, security can be set on the dial-in channel only.

The security system can be customized by individual port and applied systemwide or selectively. It has the capability of monitoring for disconnects and restoring security if the phone line drops. Security options available include name service, which requires the use of names but no passwords, or full password support, which requires passwords for access to the system.

Once the remote workstation is connected to the main system, it is treated as, and has all the capabilities of, a local station. This includes electronic mail, multitasking, etc.

CCI TERM also transfers files between the local computer and the main server. If the remote site is communicating at 9600 baud or faster and print-

ing to the remote station's workstation printer, use of RTS/CTS protocol is highly recommended. Buffering increases performance. Since CCI TERM has a 2K buffer and most terminals have only a 256-byte buffer, the addition of a hardware printer buffer between the terminal and the printer is strongly recommended.

Remote Access Quick Set-up

The following are the basic steps to set up a dial-in workstation.

1. Using the SETUP or CONFIG utility, define the serial port on the host system to be dialed into as a multiuser terminal. Use the same speed as the modem on the CCI TERM computer and Data Carrier Detected (DCD) and Request to Send/Clear to Send (RTS/CTS) protocol.

2. Use the CCIMODE utility to define that port as REMOTE — `CCIMODE X REMOTE CCPM`, where X is the port name and "CCPM" makes the definition permanent. For example:

   ```
   CCIMODE COM1 REMOTE CCPM
   ```

3. Program the modem on the host computer to auto-answer with FORCE XCOM for COM1 and COM2 or with ADUMB for addon ports. The ANSWER.BAT file supplied with CCI Concurrent DOS provides an example of using FORCE XCOM.

4. Set up dial-in security on the host computer, by creating a text file called CONTROL in the C:\CDOS\SECURITY subdirectory. Make a copy of the CONTROL.OLD file already in that subdirectory and use EDITOR to modify it for these parameters.

   ```
   MULTI
   PASSWORD
   EXCLUDE ALL
   INCLUDE Ux; (Ux = the dialin port user number)
   CLEAR BUFFERS
   HISTORY NEW
   SYSADMIN NEW
   ```

5. Use the SECURITY program to create at least one name and password. Accept all of the defaults for other parameters in SECURITY.

6. Set up the remote site with CCI TERM and a modem or a terminal with an autodial modem. Set the protocol and speed to match the dial-in port on the host system. Set SHIFT-F1 parameter to Wait on DCD and F10 to Refresh the screen.

7. Dial in, connect and go!

Setting Up the Ports On the Host Computer

The serial port on the host computer that will be dialed into should be configured as a multiuser terminal even though there are two modems between the host computer and the remote workstation. A serial port configured as a multiuser terminal can be used by a DOS communications program for dialing out. After defining the port as a multiuser terminal, use the CCIMODE utility to also define the port as REMOTE, allowing the operating system to optimally handle it. Enable the LOOKOUT utility to monitor for disconnects on ports defined as REMOTE. LOOKOUT is discussed later in this chapter.

Using the Same Modem for Dial-in and Dial-out

A single serial port can be used for both dial-in and dial-out purposes with most DOS communications programs. Configure the port in the operating system as a remote user (PC terminal or CRT). Likewise, program the modem as a dial-in modem as detailed below.

When dial-out operations are needed, invoke the DOS communications software and tell the modem to dial, connect, and complete its tasks. After telling the modem to hang-up the phone line, the serial port should be cleared with the batch statement:

```
CCIMODE COMX BAUD RESET
```

where x is the port number and baud is the dial-in communications speed. It is necessary to reset the serial port because most DOS communications programs leave the serial port in an undefined state.

Invoke the DOS communications software with a batch file that also changes communications parameters for the dial-out. Place the `CCIMODE COMX BAUD RESET` statement at the end of the batch file so that it automatically executes after exiting the DOS communications program.

Modem Configuration

There is a wide variety of modems available today, the majority claiming to be Hayes modem compatible. When setting up a remote workstation, a very important step is programming the modems on each side to "talk" to each other. Unfortunately, most 9600 baud or higher modems have command sets that vary, which results in a custom command set for programming each one. The following section includes the programming for the many common modems, and the subset of commands that most modems respond to. If the modem in use is not one of those described here, consult its manual to determine the differences and the appropriate command set for it.

Programming the Remote and Dial-in Modems

Programming of a modem can be done using ADUMB, CCI TERM, FORCE XCOM or with a TRANSIT script.

Utility	Use on ports defined as	Set handshaking	Display responses
ADUMB	AUX/PRINTER	None	Yes
CCI TERM	PC TERM, CRT	None	Yes
FORCE	PC TERM	None	No
TRANSIT	Any	None	No

NOTE: *CCI TERM must be used in ASCII emulation. FORCE can only be used with COM1 or COM2. Regardless of method, set handshaking to NONE*

If the modem being programmed is an "unknown", or has other than factory default parameters, use ADUMB or CCI TERM so that responses can be displayed. If the modem is of known manufacturer and parameters and has been successfully programmed before, use FORCE or automate the programming with a TRANSIT script (example provided later in this manual)

Multitech V32 9600 Baud Modem Initialization

These procedures initializes a Multitech V32 9600 Baud Modem. The first procedure is for enabling the modem for dial-in, and the second is to set the modem up for dial-out. There are not many differences, but they are important.

Configuring the modem connected to the host computer's serial port for dial-in access:
The following commands (without comments) need to be sent to the modem. The easiest way to do this is to set the port as an AUX port and use ADUMB to initialize the modem. Set handshaking to none. Use the AUX command to use the correct AUX port to connect the modem to. COM1 is AUX0, COM2 is AUX1, and the first Add-on Port is AUX2. OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F	Reset the modem to its factory defaults
ATQ2	No result codes on answer

ATX4	Wait for dial tone and display extended result codes
ATS0=1	Answer on first ring
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier
AT&R0	Enable CTS handshaking
AT&E1	Enable auto-reliable mode
AT&E4	RTS/CTS Handshaking enabled
AT&S1	Use DSR when ready for a call
ATS10=10	Used if call waiting is on phone line
AT$BA0	Disable baud adjust
AT#F2	Enable Fall Back/Fall Forward
AT$MB9600	Set baud rate for modem
AT#A0	Use full AutoBaud range
ATE0	Turn off command echoing. At this time, character no longer echo back
AT&W0	Store current settings in NVRAM

The switch setting on the bottom of the modem are:

8-Switch	UUUUDUDUD
4-Switch	DUDD

Internal jumpers need to be set in the modem for CTS to be independent of RTS, and DSR to be independent of DCD. Refer to chapter 9 in the Multitech modem manual. Once the modem is configured, enable RTS and CTS handshaking for the port, as that is what the modem is set up for.

Configuring the modem connected to the remote workstation's serial port for dial-out access:
The following commands (without comments) need to be sent to the modem. The easiest way to do this is to set the port as an AUX port and use ADUMB to initialize the modem. Set handshaking to none. Use the AUX command to use the AUX port that the modem is connected to. COM1 is AUX0, COM2 is AUX1, and the first Add-on Port is AUX2. OK displays after hitting Return for each command indicating that the modem understood the command.

AT&F	Reset the modem to it's factory defaults

ATQ2	No result codes on answer
ATX4	Wait for dial tone
ATS0=0	Never answer the phone
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier
AT&R0	Enable CTS handshaking
AT&E1	Enable auto-reliable mode
AT&E4	RTS/CTS Handshaking enabled
AT&S1	Use DSR when ready for a call
ATS10=10	Used if call waiting is on phone line
AT$BA0	Disable baud adjust
AT#F2	Enable Fall Back/Fall Forward
AT$MB9600	Set baud rate for modem
AT#A0	Use full AutoBaud range
AT&W0	Store current settings in NVRAM

The switch setting on the bottom of the modem are:

8-Switch	UUUUDUDUD
4-Switch	DUDD

Internal jumpers need to be set for CTS to be independent of RTS and DSR to be independent of DCD. Refer to chapter 9 in the Multitech modem manual. Once the modem is configured, enable RTS and CTS handshaking for the port, as that is what the modem is set up for.

Cardinal 9600 BPD V.42BIS Modem Initialization

This procedure initializes a Cardinal 9600 bpd V.42BIS Modem. The first procedure sets the modem for dial-in and the second sets the modem up for dial-out. There are not many differences, but they are important.

Configuring the modem connected to the host computer's serial port for dial-in access:
An OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F	Reset the modem to its factory defaults

ATN2	Result codes on dial-out only
ATY1	Enable Long Space Disconnect
ATS0=1	Answer on first ring
AT&C1	DCD tracks carrier
AT&D3	When DTR drops, drop carrier and reset modem with ATZ
AT%E1	Enable AutoTrain
AT\T15	Set inactivity timer to 15 Minutes
AT&Y0	Select configuration 0 as default
ATE0	Turn off command echoing. No further characters echo back
AT&W0	Store current settings in Configuration 0

Once the modem is configured, enable RTS and CTS handshaking for the port, as that is what the modem is now set for.

Configuring the modem connected to the remote workstation's serial port for dial-out access:
An OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F	Reset the modem to its factory defaults
ATE1	Turn on command echoing
ATN2	Result codes on dialout only
ATY1	Enable Long Space Disconnect
ATS0=0	Do not answer phone
AT&C1	DCD tracks carrier
AT&D3	When DTR drops, drop carrier and reset modem with ATZ
AT%E1	Enable AutoTrain
AT\T15	Set inactivity timer to 15 Minutes
AT&Y0	Select configuration 0 as default
AT&W0	Store current settings in Configuration 0

Once the modem is configured, enable RTS and CTS handshaking for the port since that is what the modem is now set for.

Practical Peripherals Modem Initialization

This procedure is used to initialize a Practical Peripheral, Inc. 9600 or 14400 modem. The first procedure configures the modem for dial-in and the second sets the modem for dial-out. There are not many differences, but they are important.

Connecting the modem at the host computer's serial port for dial-in access:
An OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F0	Reset the modem to its factory defaults
ATQ2	Result codes on dial-out only
ATW0	Do not return negotiation messages
ATS0=1	Answer on first ring
AT&C1	DCD tracks carrier
AT&D3	When DTR drops, drop carrier, reset modem
ATY1	Enable long space disconnect
AT&Y0	Select profile 0 on power-up
ATE0	Turn off command echoing. No more characters echo back
AT&W0	Store current settings in NVRAM

Once the modem is configured, enable RTS and CTS handshaking for the port since that is what the modem is now set for.

Configuring the modem connected to the remote workstation's serial port for dial-out access:
An OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F0	Reset the modem to its factory defaults
ATQ2	Result codes on dial-out only
ATW1	Return negotiation messages
ATS0=0	Do not answer phone
AT&C1	DCD tracks carrier
AT&D3	When DTR drops, drop carrier and reset modem

ATY1	Enable long space disconnect
AT&Y0	Select profile 0 on power-up
ATE0	Turn off command echoing. No more characters echo back
AT&W0	Store current settings in NVRAM

Once the modem is configured, enable RTS and CTS handshaking for the port since that is what the modem is now set for.

Zoom VFX V32bis Modem Initialization

This procedure initializes a 9600 or 14400 Zoom VFX Modem. The first procedure configures the modem for dial-in and the second for dial-out. There are not many differences, but they are important.

Connecting the modem to the host computer's serial port for dial-in access:
An OK displays after hitting RETURN for each command indication that the modem understood the command.

AT&F0	Reset the modem to its factory defaults
ATQ2	Result codes on dial-out only
ATW0	Do not return negotiation messages
ATS0=1	Answer on first ring
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier, reset modem
ATY1	Enable long space disconnect
AT&Y0	Select profile 0 on power-up
ATE0	Turn off command echoing. No more characters echo
AT&W0	Store current settings in NVRAM

Once the modem is configured, enable RTS and CTS handshaking for the port since that is what the modem is now set for.

Configuring the modem connected to the remote workstation's serial port for dial-out access:
An OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F0	Reset the modem to its factory defaults

ATQ2	Result codes on dial-out only
ATW1	Return negotiation messages
ATS0=0	Do not answer phone
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier and reset modem
ATY1	Enable long space disconnect
AT&Y0	Select profile 0 on power-up
ATE0	Turn off command echoing. No more characters echo back
AT&W0	Store current settings in NVRAM

Once the modem is configured, enable RTS and CTS handshaking for the port since that is what the modem is now set for.

U.S. Robotics Courier Modem Initialization

This procedure initializes a U.S. Robotics Courier Modem. The first procedure enables the modem for dial-in and the second for dial-out. There are not many differences, but they are important.

Configuring the modem connected to the host computer's serial port for dial-in access:

The following commands (without comments) need to be sent to the modem. The easiest way to do this is to set the port as an AUX port and use ADUMB to initialize the modem. Set handshaking to none. Use the AUX command for the AUX port that the modem is connected to. COM1 is AUX0, COM2 is AUX1, and the first Add-on Port is AUX2. OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F	Reset the modem to its factory defaults
ATQ2	Result codes on dial-out only
ATS0=1	Answer on first ring
AT&A3	More verbose statuses available
AT&B1	Keep talking to modem at established baud
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier

AT&H1	Use CTS handshaking
AT&R2	Use RTS handshaking
AT&S1	Use DSR when ready for a call
ATS10=10	Used if call waiting is on phone line
ATS13=1	Used to reset modem when DTR drops
ATE0	Turn off command echoing.
AT&W	Store current settings in NVRAM

Once modem is configured, enable RTS and CTS handshaking for the port since that is what the modem is set up for.

Configuring the modem connected to the remote workstation's serial port for dial-out access:
The following commands (without comments) need to be sent to the modem. The easiest way to do this is to set the port as an AUX port and use ADUMB to initialize the modem. Set handshaking to none. Use the AUX command to set the AUX port that the modem is connected to. COM1 is AUX0, COM2 is AUX1, and the first Add-on Port is AUX2. OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F	Reset the modem to its factory defaults
ATQ2	Result codes on dialout only
ATS0=1	Answer on first ring
AT&A3	More verbose statuses available
AT&B1	Keep talking to modem at established baud
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier
AT&H1	Use CTS handshaking
AT&R2	Use RTS handshaking
AT&S1	Use DSR when ready for a call
ATS10=10	Used if call waiting is on phone line
ATS13=1	Used to reset modem when DTR drops
ATE0	Turn off command echoing. No more characters echo back

AT&W	Store current settings in NVRAM

Once the modem is configured, enable RTS and CTS handshaking for the port since that is what the modem up is set for.

U.S. Robotics Sportster Modem Initialization

This procedure initializes a 9600 or 14400 U.S. Robotics Sportster Modem. The first procedure configures the modem for dial-in and the second for dial-out. There are not many differences, but they are important.

Configuring the modem connected to the host computer's serial port for dial-in access:
An OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F	Reset the modem to its factory defaults
ATQ2	Result codes on dial-out only
ATS0=1	Answer on first ring
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier
AT&H1	Use CTS handshaking
AT&R2	Use RTS handshaking
AT&S1	Use DSR when ready for a call
ATS10=10	Used if call waiting is on phone line
ATS13=1	Used to reset modem when DTR drops
ATE0	Turn off command echoing. No more characters echo back
AT&W	Store current settings in NVRAM

Once the modem is configured, enable RTS and CTS handshaking for the port since the modem is set for this.

Configuring the modem connected to the remote workstation's serial port for dial-out access:
An OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F	Reset the modem to its factory defaults
ATQ2	Result codes on dial-out only

ATS0=0	Do not answer phone
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier
AT&H1	Use CTS handshaking
AT&R2	Use RTS handshaking
AT&S1	Use DSR when ready for a call
ATS10=10	Used if call waiting is on phone line
ATS13=1	Used to reset modem when DTR drops
AT&W	Store current settings in NVRAM

Once the modem is configured, enable RTS and CTS handshaking for the port, as modem is set for this.

INTEL EX Modem Initialization

This procedure initializes an INTEL 9600 or 14400 EX Modem. The first procedure enables the modem for dial-in and the second for dial-out. There are not many differences, but they are important.

Configuring the modem connected at the host computer's serial port for dial-in access:
The following commands (without comments) need to be sent to the modem. The easiest way to do this is to set the port as an AUX port and use ADUMB to initialize the modem. Set handshaking to none. Use the AUX command to set up the modem's AUX port. COM1 is AUX0, COM2 is AUX1, and the first Add-on Port is AUX2. An OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F	Reset the modem to its factory defaults
ATS0=1	Answer on first ring
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier
AT&R0	Use RTS handshaking
AT&S0	DSR ON always
AT\C1	Enable Auto-Reliable Buffer
AT\N3	Auto-Reliable Mode enabled

AT\Q3	Enable RTS/CTS handshaking
ATS10=100	Used if call waiting is on phone line
ARS25=25	Reset DTR detect timer
AT&Y0	Use profile 0 as default
ATQ1	No Result codes
ATE0	Turn off command echoing. No more characters echo back
AT&W0	Store current settings in NVRAM

Once the modem is configured, enable RTS and CTS handshaking for the port since that is what the modem is set for.

Configuring the modem connected to the remote workstation's serial port for dial-out access:

The following commands (without comments) need to be sent to the modem. The easiest way to do this is to set the port as an AUX port and use ADUMB to initialize the modem. Set handshaking to none. Use the AUX command to set the AUX port that the modem is connected to. COM1 is AUX0, COM2 is AUX1, and the first Add-on Port is AUX2. An OK displays after hitting RETURN for each command indicating that the modem understood the command.

AT&F	Reset the modem to its factory defaults
ATS0=0	Never answer the phone
AT&C1	DCD tracks carrier
AT&D2	When DTR drops, drop carrier
AT&R0	Use RTS handshaking
AT&S0	DSR ON always
AT\C1	Enable Auto-Reliable Buffer
AT\N3	Auto-Reliable Mode enabled
AT\Q3	Enable RTS/CTS handshaking
ATS10=100	Used if call waiting is on phone line
ARS25=25	Reset DTR detect timer
AT&Y0	Use profile 0 as default
ATQ0	Show Result codes

AT&W0	Store current settings in NVRAM

Once the modem is configured, enable RTS and CTS handshaking for the port since that is what the modem is set for.

Most Commonly Used AT Commands For Modems

For the modem without a manual, the commands in this section may enable it to be successfully programmed. Most modems respond to the common set of basic programming commands listed below. When any of these commands are used, either an ERROR or an OK message returns from the modem. ERROR means that the modem does not support this particular command. OK indicates that the modem accepts and is responding to this command.

ATZ	Resets modem.
ATF1	Selects full duplex operation (simultaneous bi-directional transmission). Default value for all modems
ATE1	Causes commands to be echoed on the screen.
AT&C1	Causes DCD (Data Carrier Detected) to track the received carrier signal.
AT&D2	Causes the modem to go back on-hook and disable auto-answer on the ON-to-OFF transition of DTR (Data Terminal Ready). Auto-answer reenables (if it was disabled) when DTR is turned on.
AT&L0	Selects standard telephone line. (Default for all modems)
AT&L1	Selects leased line.
AT&E1	Puts the modem into the auto reliable mode known as MNP (hardware error correction).
ATS0=1	Sets the modem to auto-answer the phone after 1 ring. This can be set from 1 to 255 rings.
ATS7=30	Sets the modem to time out after 30 seconds if no carrier is found. This can be set from 1 to 255 seconds.
AT&W0	Saves all of the AT commands into non volatile RAM memory if it exists or else returns an error message.

Troubleshooting Modem Problems

Following are some guidelines for troubleshooting problems that may arise. Throughout this section the following syntax will be used:

☐ *A symptom or problem scenario.*

? Explanation of the problem.

S Solution – how to solve the problem.

H Hint – how to identify the problem.

T Tip – how to improve performance.

☐ *The modem does not auto answer under DOS.*

? The modem may need special configuration for auto answer.

S To configure the modem to auto answer under DOS:

1. Get PATH to recognize the MODE command

2. At the command prompt, or in the AUTOEXEC.BAT file, put:

```
C>MODE COM1:baudrate
C>ECHO ATZ    >COM1
C>ECHO ATS0=1 >COM1
C>ECHO AT&W0  >COM1
```

The ATZ string resets the modem, the S0=1 string changes S-register zero to answer the phone on 1 ring, and the &W0 string saves the changed register in the modem. **This must all be in capital letters.** If this is in the AUTOEXEC.BAT file, it must appear before a LOADSYS.

☐ *The modem does not auto answer under CCI Concurrent DOS.*

? The modem may need a special configuration for auto answer. ANSWER.BAT, a batch file included with the operating system, uses FORCE and operates on COM ports defined as multiuser terminals. It includes the line FORCE XCOM ATS0=1.

1. CCIMODE COMX T NONE
2. CAPS LOCK
3. Copy CCPM.SYS to CCPM.OLD
4. Run SETUP and configure the channel as printer AUX
5. Reboot
6. AUX = X
7. ADUMB
   ATZ
   ATSO=1
   AT&WO

CTRL C
8. Copy CCPM.OLD to CCPM.SYS
9. Reboot

☐ *Modem answers phone, station has no prompt.*

? This is caused by incorrect protocol.

S Reset the protocol and reboot the computer while remaining connected. A prompt should now be displayed.

☐ *Remote printer dead or shows garbage.*

? This sounds like all of the necessary pins are not connected.

S Confirm that the correct AUX port is being used with the AUX command.

S Check to be sure that echo is set to on

S Check to see if pin 6 on the computer is connected to pin 8 of the modem. If it is, then look at the problem as a workstation printer problem rather than a modem-related problem.

☐ *The printer (or some other printer) starts printing as soon as a connect is achieved.*

? This is caused by noise on the line. Some portion is interpreted as a ^P, causing the default printer to begin printing the contents of the screen.

S Assign the remote channel to a "phantom" printer so that the ^P's are harmless. After login, reassign the station to the printer to be used with the appropriate NAME.BAT file. Enable SECURITY and LOOKOUT.

☐ *The modem completes a dial-out under DOS but not under CCI Concurrent DOS.*

? The modem port must not be owned by any other process in order to dial-out on it. The DOS program may be accessing the hardware directly and reprogramming the communications parameters. To dial-out under CCI Concurrent DOS, the communication parameters must match what the operating system has for parameters for that serial port.

S Check to see if CSPOOL is running. If so, check the CONFIG.CPL file to see if the printer number represented by the AUX port (3 for COM1, 4 for COM2) is included in the NOT_PRINTER parameter. If not, include it.

S Check the baud rate, parity, and protocol for the port in use.

Monitoring Remote Channel Status

The CCI Concurrent DOS LOOKOUT program monitors the serial channels' status on ports that are defined as remote and takes an action (such as disconnecting the modem) based on that status. This program can maintain and restore security when the phone line drops. It can also be used to terminate a session based on inactivity. REMOTE is the program that defines the parameters used by LOOKOUT.

LOOKOUT

The LOOKOUT program runs in the background and monitors remote access channels. CCIMODE must be used to identify a channel as being remote before starting LOOKOUT. When a disconnect condition on a remote channel is detected, LOOKOUT takes action to return the terminal to a secure state, usually by terminating the programs running on the terminal and allowing the Concurrent Controls Inc. Security shell to regain control of the console. The actual parameters used by LOOKOUT are defined by the REMOTE program. The LOOKOUT program has the following command line syntax:

```
LOOKOUT [/H] [/S] [/Q] [/I]
```

where:

/H	Help — displays a help screen
/S	Stop — stops LOOKOUT
/Q	Quiet — starts LOOKOUT without any messages
/I	Inquire — displays if LOOKOUT is running

Enter LOOKOUT with no parameters to start the program running in the background. To start LOOKOUT automatically when the system boots put the LOOKOUT command in the CCIBOOT.BAT or STARTxxy.BAT batch files.

REMOTE

The REMOTE program defines the operating parameters for LOOKOUT. The options for LOOKOUT fall into two categories: how LOOKOUT determines a disconnect condition has occurred and what LOOKOUT does when a disconnect is detected. The REMOTE program has the following command line syntax:

```
REMOTE [/H] [/n] [Override]
```

where:

/H	Help — displays a help screen
/n	Skips to LOOKOUT option number n
Override	Overrides revision level check.

The LOOKOUT options, their /n option numbers, and their default values are:

1. Lookout Handshaking — Yes
2. Lookout Inactivity — No
3. Lookout Send Character — No
4. Lookout Terminate Foreground — Yes
5. Lookout Terminate Background — Yes
6. Lookout Terminate Non-System — Yes
7. Lookout Inactivity Time — 300 seconds (5 min)
8. Lookout Handshaking Time — 20 ticks (1/3 sec)
9. Lookout Delay — 20 ticks (1/3 sec)
10. Lookout Handshaking Mask — 80h (DCD line)
11. Lookout Character — 03h (CTRL-C)
12. Lookout Set Disconnect — Yes
13. Lookout Abort Process — Yes
14. Lookout Drop DTR — Yes
15. TMP Login Inactivity Time — 0 seconds (disabled)

As shipped, REMOTE has LOOKOUT terminate all processes running on a terminal if the data carrier detect (DCD) handshaking line is dropped for at least 1/3 of a second. These settings work for modem channels and handles the situation when a modem line disconnects.

The REMOTE program scrolls through the LOOKOUT options, asking for a value to be entered for each option. Skip to a specific option by specifying an option number on the REMOTE command line. For example, to change the Lookout Delay (option number 9), use the command:

```
REMOTE /9
```

REMOTE must be run from the directory where the CCPM.SYS system file is located, normally the root directory.

The next section describes each option in REMOTE in greater detail.

LOOKOUT Options Set by REMOTE

1. Lookout Handshaking

This option determines if LOOKOUT monitors hardware handshaking lines for a disconnect condition. If this option is Yes, and one of the handshaking lines specified in Lookout Handshaking Mask drops for the time specified in Lookout Handshaking Time, a disconnect has occurred. If this option is No, LOOKOUT does not monitor the handshaking lines and ignores the Lookout Handshaking Mask and Lookout Handshaking Time fields.

2. Lookout Inactivity

This option determines if LOOKOUT monitors keyboard activity for a disconnect condition. If this option is Yes, and the remote keyboard is inactive for the time specified in Lookout Inactivity Time, a disconnect has occurred. If

this option is No, LOOKOUT does not monitor keyboard activity and ignores the Lookout Inactivity Time field.

3. Lookout Send Character

This option determines if LOOKOUT terminates a disconnected task by sending it a character. This feature sometimes allows the graceful exit of a task that may not terminate properly if LOOKOUT simply terminated it. If this option is Yes, and a disconnect condition occurs, LOOKOUT sends the character defined in the Lookout Character field to the disconnected task. If this option is No, LOOKOUT does not send characters to disconnected tasks and ignores the Lookout Character field.

4. Lookout Terminate Foreground

This Yes or No option permits LOOKOUT to terminate the foreground task running on a disconnected console. This option is usually Yes. If this option is No, foreground tasks must be terminated by some other process.

5. Lookout Terminate Background

This Yes or No option permits LOOKOUT to terminate any background tasks running on a disconnected console. This option is usually Yes. If this option is No, background tasks must be terminated by some other process.

6. Lookout Terminate Non-System

This option allows LOOKOUT to continue to terminate tasks running on a disconnected console until LOOKOUT runs into a system process (such as the CCI Security shell). Some programs include a "shell to DOS" function which allows you to run a second program while inside the first program. The Lookout Terminate Non-System option is usually set to Yes to allow LOOKOUT to terminate all such "user" processes.

7. Lookout Inactivity Time

When LOOKOUT is checking remote channels for keyboard activity, this is the number of seconds a keyboard has to remain inactive before LOOKOUT decides the channel has disconnected. This field can range from 0 to 65,535 seconds (18 hours). The default is 300 seconds (5 minutes).

8. Lookout Handshaking Time

When LOOKOUT is monitoring handshaking lines, this is the number of system ticks a handshaking line has to drop before LOOKOUT decides the channel has disconnected. A system tick is 1/60 of a second (about 16 milliseconds). The default value is 20 ticks or 1/3 of a second. This field should be equal to or greater than the Lookout Delay field described below.

9. Lookout Delay

This is the number of system ticks LOOKOUT delays between checking remote ports for disconnect conditions. A larger number reduces system overhead, a lower number causes quicker detection of a disconnect. The default value is 20 ticks, or 1/3 of a second. This field should not be greater than the Lookout Handshaking Time field described above.

10. Lookout Handshaking Mask

LOOKOUT uses this bit pattern to determine which hardware handshaking lines to monitor for connect or disconnect conditions. The bit pattern is specified in hexadecimal. The bits used are:
80h: Use Data Carrier Detect (DCD)
10h: Use Clear To Send (CTS)
20h: Use Data Set Ready (DSR)

To select more than one line, add the values together. For example, use 90h for DCD/CTS, A0h for DCD/DSR, 30h for CTS/DSR, or B0h to select all lines. The default value is 80h which monitors the DCD line for modem disconnects.

11. Lookout Character

When the Lookout Send Character field is Yes, this is the hexadecimal character sent to a running task when a disconnect condition has been detected. Some common values for this field are:
03: CTRL-C
1B: ESCAPE
59: Y (for Yes)
4E: N (for No)

12. Lookout Set Disconnect

Determines whether LOOKOUT sets the channel disconnect flag upon detecting a disconnect. This field is usually Yes.

13. Lookout Abort Process

This Yes or No field determines whether LOOKOUT terminates tasks running on disconnected channels. This option is usually Yes to allow LOOKOUT to simply terminate processes. If LOOKOUT uses Lookout Send Character to terminate processes, set this field to No.

14. Lookout Drop DTR

If CCIMODE HANGUP, BYE, or the CCI Security shell ask LOOKOUT to disconnect, and this field is set to Yes, LOOKOUT drops the DTR modem control line to make the modem hang up.

See Chapter 2, Utility Reference, in the *CCI Concurrent DOS Utilities and Troubleshooting Guide* for more information on the BYE and CCIMODE utilities. See

Chapter 4, Security System, for information on the CONTROL file option BAD-NAME-DISCONNECT.

15. TMP Login Inactivity Time This is the number of seconds to wait at the LOGIN prompt after a modem indicates a connection (raises DTR) before LOOKOUT will drop DTR (to disconnect the line) if no keystrokes are received. The default of 0 disables this option, otherwise a value of 60 to 65535 seconds (18 hours) is allowed.

REMOTE Examples

The REMOTE program selects a variety of remote access options for different situations. The following examples assume that CCIMODE REMOTE has been used to flag the channel as a remote access channel and LOOKOUT is running in the background.

☐ *To control phone charges, remote users should be logged off if they do not touch the keyboard for 10 minutes.*

S Set Lookout Inactivity to YES and Lookout Inactivity Time to 600 seconds. When LOOKOUT detects the remote keyboard has been inactive for 10 minutes, LOOKOUT terminates all foreground and background non-system processes and logouts the user.

☐ *A control line other than DCD (Data Carrier Detected) is needed to indicate a disconnect.*

S Change the Lookout Handshaking Mask to select the control line in use.

☐ *A batch file needs to continue executing after the current process terminates.*

S Set Lookout Send Character to YES, Lookout Character to 4E (an ASCII "N"), and Lookout Terminate Non-System to NO. When LOOKOUT detects a disconnect condition, it terminates the current program and sends the answer "N" to the `Halt batch file (Y/N) ?` question.

☐ *Since the application requires more than a single character to terminate, the operating system should NOT terminate it, leaving that operation for a batch file or manual entry.*

S Set Lookout Abort Process to NO so LOOKOUT does not terminate processes directly and instead have LOOKOUT send characters to the disconnected task. This feature is for software developers who know how to terminate their own programs. It is usually difficult to determine what character sequences are guaranteed to terminate an "off the shelf" application under all circumstances. This feature allows an orderly shutdown of a known program.

If the disconnected task terminates based on a single character, say ESCAPE, set Lookout Send Character to YES and Lookout Character to 1B (an ASCII ESCAPE). If the program terminates upon receipt of a single character, set Lookout Terminate Non-System to NO. If the program must receive the terminate character multiple times, set Lookout Terminate Non-System to YES so LOOKOUT continues to send the character until the program actually terminates. This could be used if a single character steps the user back through multiple levels of menus before finally exiting.

☐ *The application needs more than a single character to terminate.*

S If a program requires a more complicated sequence of characters to terminate, a background batch file can be used with FORCE sending keystrokes to the foreground application. Set the following LOOKOUT options.

Lookout Terminate Foreground — No
Lookout Terminate Background — Yes.
Lookout Send Character — Yes
Lookout Character — 4E (an ASCII "N")
Lookout Terminate Non-System — No
Lookout Abort Process — Yes

The result is that the foreground application continues execution and all termination logic occurs on the background sessions windows when a disconnect occurs. The background batch file could be based on the following:

```
:LOOP
FLAGWAIT 87
FORCE - [B]X~
GOTO LOOP
```

When this batch file runs on a background terminal, it executes a FLAGWAIT on flag number 87 (the flag number used may differ, see WHATFLAG to determine an unused flag number on the system). The idea is to use a flag number that never FLAGSETs by any other process so the batch file essentially halts until FLAGWAIT is terminated by LOOKOUT. After LOOKOUT terminates FLAGWAIT, LOOKOUT sends an ASCII "N" character to the background window to answer the `Halt Batch Process (Y/N)` ? question. The batch file then continues and uses FORCE to send characters to the foreground process. In this example FORCE sends the character sequence CTRL-K-X to terminate WordStar running in a foreground window.

Remote Support

Both remote access and remote support functions are built into every CCI Concurrent DOS system. Once the remote access stations have connected, the remote support functions included with CCI Concurrent DOS — TAKEOVER and TRANSIT — can be used to do system setup, maintenance and diagnostics, uploading and downloading of files, operating system and application training.

TAKEOVER allows the "sharing" of displays between two or more users on the system, so that all users see the same screen displays. It can also be used to view the input to and output of other system resources, including printers and aux ports. The TAKEOVER program can be used for diagnostic, training, or monitoring purposes. It should be be password protected to prevent unauthorized access.

TAKEOVER has the following command line syntax:

```
TAKEOVER [/H] [device name] [device number][options]
```

where:/H displays help text and examples.

TAKEOVER can be best understood if considered in terms of who (the user or device) is being taken over, how they are being taken over, and how the session is ended. When the term "sender" is used, it refers to the user initiating TAKEOVER, and "receiver" is the user or device being taken over.

The User/Device that is Being Taken Over

The TAKEOVER default is to take over users and monitor their output.

■ *Example: Take over session three and watch their display.*

```
TAKEOVER 3
```

U — user number

The user number corresponds to the session number. When a session number is specified, TAKEOVER defaults to displaying the keyboard and display of the current foreground process on that physical console. For example, if U2 is the foreground process on the main console when TAKEOVER U3 is executed on station 5, U2 is the session displayed to the user on station 5. TAKEOVER is physical console oriented, when one session on a console is taken over, all sessions on that console can be viewed.

P – printer number.

The parallel ports are printer numbers 0, 1, and 2; COM1 is printer 3, COM2 is printer 4, etc. The TAKEOVER default is to monitor the system output to printer.

A — aux port or communications port number.
COM1=AUX0, COM2=AUX1, etc. When monitoring AUX ports, the TAKEOVER default is to monitor input from the specified AUX port.

S — workstation or physical console number.
The host console is station 0. When taking over a station, the TAKEOVER default is to control keyboard and display with the current foreground process active.

R — remote, taking over a PCTERM station through an AUX port.
This allows remote support to be done through a AUX channel that is set up for dial-out communications. The remote option gives control of a PCTERM terminal to the specified AUX port. This command is executed from a PC terminal (or a port defined as PTX) on the remote system before the dial-in is made. When the dial-in connection is made, the PC terminal is dedicated to the AUX port until the completion of the dial-in operation.

When using the remote option, it is recommended that the serial port hardware be reset to ensure a clean connection. This is done with `CCIMODE COMx BAUD RESET`. The ANSWER.BAT batch file included with each CCI Concurrent DOS system provides an example of using CCIMODE.

- *Example 1: Allow remote access via COM1 (AUX 0). Run the following from a PC terminal, FORCE it to a port defined as PTX, or use the SCHEDULER to execute it automatically.*

  ```
  TAKEOVER A0 R
  ```

How That User or Device is Being Taken Over

The default is for both keyboards (sender and receiver) to be active and for device output to be displayed.

- *Example: Takeover user 9 with both keyboards active.*

  ```
  TAKEOVER 9
  ```

M — Monitor Mode
This mode allows the sender to monitor or watch the display and output (keyboard activity if a user) of the receiver, with the sender's keyboard inactive.

- *Example 1: Monitor the screen of user 5*

  ```
  TAKEOVER U5 M
  ```

- *Example 2: Monitor the output of printer 0.*

  ```
  TAKEOVER P0 M
  ```

L — Lock the keyboard of the specified user.
Only to be used with the T and R options.

```
TAKEOVER U9 L T
```

■ *Example: Takeover user 9's terminal and lockout their keyboard.*

```
TAKEOVER 9 L T
```

B — Bitbucket the output of the specified user.
Only to be used with the T and R options.

■ *Example: Takeover user 9's turned-off terminal to exit the program(s) left running when the terminal was turned off. Use the Bitbucket option because the terminal is turned off, making any hardware handshaking inactive.*

```
TAKEOVER 9 B R
```

I — Monitor Device Input
This mode only monitors or watches the input of the selected device.

■ *Example 1: Monitor COM1 port input.*

```
TAKEOVER A0 I
```

Q — Takeover Quiet mode.
Ready message is not displayed. Use this option to start TAKEOVER in a batch file.

Ending the TAKEOVER session

To terminate a TAKEOVER session normally, press the ALT-SYSREQ key on the Host console and PCTERM terminals, or the ' key on ASCII terminals. If a TAKEOVER session, once underway, is aborted — not ended normally — executing the next option, X, is critical.

■ *Example: Extract the TAKEOVER hooks caused by takeover of user 9.*

```
TAKEOVER U9 X
```

TRANSIT

Along with full remote access and remote support capabilities, the CCI Concurrent DOS operating system includes TRANSIT, a utility to transfer files within a multiuser system (to and from workstations and the main computer system), or with other computer systems (over a network or connected to other computers). TRANSIT has a script language that automates file transfers and other dial-in or related functions, including the programming and dialing of modems. Those scripts can be combined with batch files and the Scheduler to achieve automatic execution at pre-specified times, dates or other conditions.

TRANSIT runs under CCI Concurrent DOS or under DOS. TRANSIT uses either XMODEM or YMODEM protocol for error-free transfer of files between two computers connected by their serial ports. The two ports can be directly connected (a local connection) or they can be connected through two modems and a telephone line (a remote connection). When transferring files locally, data transfer rates can be as fast as the serial ports are configured —up to 115 KB. When transferring through modems, the file transfer rate will depend on the speed of the modem, typically 2400 or 9600 baud.

Under CCI Concurrent DOS, TRANSIT uses either a COM port or an AUX port on a standard or intelligent serial port board. Under DOS, TRANSIT uses COM1, COM2, COM3, or COM4.

TRANSIT can be used to exchange files with CCI TERM or with any other communications program that supports the XMODEM or YMODEM protocols. TRANSIT can be programmed to automatically perform a series of commands by using a TRANSIT script file. Although TRANSIT is limited to the sending and receiving of files when used from the command line, far more complex operations can be performed through TRANSIT scripts. For example, a TRANSIT script can automatically dial into a remote system, download a series of files, and then logout.

TRANSIT can be used with the Scheduler to automatically perform file transfers on a given day and time. This is useful to perform unattended file transfers in the evening, when phone rates can be lower and the system is otherwise inactive.

TRANSIT Basics

File transfer requires a stable serial connection between two computers and file transfer programs using the same file transfer protocol running on each end. One computer runs a file transfer program in "send" mode while the other computer runs in "receive" mode. Once the two programs establish contact with each other (handshaking) they begin the file transfer by

breaking the file into pieces (blocks) and sending/verifying a block at a time.

A simple example is two computers directly connected with a serial cable running TRANSIT on each end. One computer executes TRANSIT in send mode and the other computer executes TRANSIT in receive mode.

A more complicated example could involve using CCI TERM or a TRANSIT script to dial into a remote CCI Concurrent DOS system. Once successfully logged into the remote system and at a command prompt, start TRANSIT on the remote computer in send or receive mode. The remote TRANSIT waits to establish the initial file transfer handshake. From the local side, enter CCI TERM's file transfer menu or have the TRANSIT script begin the file transfer. Within a few moments the local and remote programs handshake and begin the actual file transfer. When the transfer completes, the local user stays connected to the remote system and both systems are returned to the command prompt, ready to continue.

Performance Considerations

The type of communications port used (COM1, COM2, standard board, or intelligent board) and the level of activity on the CCI Concurrent DOS system determines TRANSIT's maximum effective data transfer rate. Intelligent boards give better performance than COM ports or standard boards because they buffer data in their on-board memory. With intelligent boards, it is possible to send and receive up to 115 Kbps.

Standard serial ports (with 16450 communications chips, including COM1, COM2, and non-intelligent serial boards) usually receive files up to 38.4 Kbps and send files up to 115 Kbps. System load affects serial transfer performance, resulting in lower actual buad rates.

The level of system activity impacts the performance of TRANSIT, and TRANSIT impacts the performance of the rest of the system. The highest level of performance comes by running TRANSIT under DOS or as a single task under CCI Concurrent DOS. If TRANSIT has an unacceptable performance impact on other programs under CCI Concurrent DOS, use the Scheduler to transfer files during times of non-peak activity or use the TRANSIT PDELAY option to slow TRANSIT down.

TRANSIT Command Line Usage

The TRANSIT utility has two command line formats: one when running a TRANSIT script, and one when running TRANSIT directly from the command line. The command line format when running a TRANSIT script file is:

```
TRANSIT [/HS] @[ScriptFile] [/V] [Parameters]
```

/HS Displays help on the TRANSIT script language.

ScriptFile The script file name. The "@" character must precede the script file name.

/V An optional parameter to verify script files for proper syntax. Each command in the script file is processed but not actually executed. This ensures that there are no obvious syntax errors in the script file. The /V option must appear immediately after the script file name.

Parameters Up to nine script file replacement variables used to pass parameters from the command line to the script file. Command line parameters can be accessed inside the script file by using the %N script command, where N is a number between 1 and 9.

The command line format when running TRANSIT directly from the command line is:

```
TRANSIT [/H] [Port] [Protocol] [Send|Rec]
[Filename] [Overwrite] [Timeout] [PDelay]
```

where:

/H Displays help text and examples.

Port Specifies the serial port TRANSIT uses. When running TRANSIT under DOS, valid entries are COM1, COM2, COM3 (IRQ 5, port 3E8h), or COM4 (IRQ 2, port 2E8h). When running TRANSIT under CCI Concurrent DOS, valid entries are COM1, COM2, or AUX 0 to AUX 99. Run PRINTMAP /A to display a list of valid AUX port numbers for the system.

If the port is not specified on a CCI Concurrent DOS system, TRANSIT uses the current console port. For example, it is not necessary to specify a port when executing TRANSIT on a remote system while running CCI TERM because file transfers occur through the current console port.

Protocol Specifies the file transfer protocol. Use XMODEM (or X) for XMODEM 128-byte/block protocol and YMODEM (or Y) for YMODEM 1024-byte/block protocol. YMODEM is usually more efficient than XMODEM protocol, but XMODEM gives better performance on noisy connections.

If protocol is not specified, protocol defaults to XMODEM. When using XMODEM, TRANSIT first tries to communicate using XMODEM-CRC protocol. If that fails,

TRANSIT automatically switches to XMODEM–CHECKSUM protocol.

Send|Rec Specifies if a file is to be sent or received. Use S (or SEND) to send a file and R (or REC or RECEIVE) to receive a file.

Filename Specifies the path and/or filename to send or receive.

Overwrite O (or OVERWRITE) is an optional parameter for the RECEIVE option. If using RECEIVE without OVERWRITE and a filename is supplied that already exists, TRANSIT asks:

```
Specified file exists. Overwrite (Y/N) ?
```

If using RECEIVE with OVERWRITE, TRANSIT replaces any existing file with the received file without asking. This option is useful for running TRANSIT in a batch file when the batch file is not to be interrupted by user input.

Timeout T (or TIMEOUT) is an optional parameter that overrides the default timeouts for retrying and aborting a transfer. The T option is followed by the number of seconds (between 0 and 255) to timeout. If TIMEOUT is not specified, 30 seconds is used for receive and 60 seconds is used for send.

Pdelay P (or PDELAY) is an optional parameter for use under CCI Concurrent DOS only and has no effect under DOS. This option is the number of system ticks TRANSIT waits between processing characters. A system tick is 1/60 of a second (about 16 milliseconds). The P option is followed by the number of ticks (between 0 and 255) to delay. The default is 1 tick.

PDELAY is useful if running other programs under CCI Concurrent DOS while doing a file transfer. Since TRANSIT is a CPU-intensive program, other programs may slow down unacceptably during a file transfer. Increasing PDELAY slows down TRANSIT but increases performance of other programs.

- *Example 1: Receive a file through the current console and call it TEST.DAT. Use the default XMODEM protocol and the default timeout value (30 seconds).*

  ```
  TRANSIT R TEST.DAT
  ```

- *Example 2: Send the file TEST.DAT through the COM1 port using XMODEM protocol.*

  ```
  TRANSIT COM1 X S TEST.DAT
  ```

- *Example 3: Receive the file TEST.DAT from the AUX0 serial port. Use YMODEM protocol, a timeout of 10 seconds and overwrite the file TEST.DAT if it already exists. The AUX 0 TRANSIT option is not available under DOS.*

  ```
  TRANSIT AUX 0 Y R TEST.DAT OVERWRITE T 10
  ```

TRANSIT Script Language

TRANSIT can be controlled by a sequence of commands in a TRANSIT script file as well as performing basic file transfers from the command line. A script file is an ASCII text file containing one TRANSIT script command per line. To execute a script file, place the script filename preceded by an "@" character on the command line. For example:

```
TRANSIT @SCRIPT
```

executes the commands in the script file SCRIPT. The following section described the TRANSIT script commands. Some script commands can be abbreviated. For example, RECEIVE, REC, or R all receive a file.

AUX

Syntax: `AUX #`

The AUX command tells TRANSIT to communicate through AUX port number #. Valid entries are AUX 0 for COM1, AUX 1 for COM2, up through AUX 99. The AUX command is not valid under DOS. When writing a script that runs under both DOS and CCI Concurrent DOS, use the IF CCI syntax to test so that the AUX # command only executes under CDOS.

```
IF CCI AUX 1  ; use aux port 1 (COM2)
```

AUX # (or COM# or CONSOLE) must be used to specify a communications port before beginning a file transfer.

CHAIN

Syntax: `CHAIN scriptfile`

The CHAIN command transfers control to another TRANSIT script file. This command is used to execute multiple TRANSIT scripts during a single TRANSIT session. A CHAINed script file does not return to the original script file. CHAIN can be used for executing commands in a "cleanup" script when an error occurs:

```
SEND MYFILE
IF ERROR CHAIN @BADSEND
```

COM

Syntax: `COM #`

The COM command tells TRANSIT to communicate through COM port number #. Valid COM port numbers under CCI Concurrent DOS are COM1

or COM2. Valid COM port numbers under DOS are COM1 through COM4. To se COM3 under DOS, configure the port for IRQ5. To use COM4 under DOS, configure the port for IRQ2. Use IF DOS to test so those ports are not selected under CDOS.

```
IF DOS COM 3    ; use COM3
```

COM# (or AUX # or CONSOLE) must be used to specify a communications port before beginning a file transfer.

CONSOLE

Syntax: `CONSOLE`

The CONSOLE command tells TRANSIT to communicate through the current console port. CONSOLE is the default communications channel under CCI Concurrent DOS. The CONSOLE command is not valid under DOS.

```
IF DOS COM1       ;use COM1 under DOS
IF CCI CONSOLE    ;use console under CCI
```

The CONSOLE command is used when using CCI TERM or a TRANSIT script to dial into a remote system's terminal port. Have the remote system execute a TRANSIT script containing the CONSOLE command so the remote TRANSIT communicates with the local CCI TERM or TRANSIT script through the terminal port.

CONSOLE (or AUX # or COM#) must be used to specify a communications port before beginning a file transfer.

DIAL

Syntax: `DIAL string`

The DIAL (or D) command sends a dialing string to the modem. The DIAL command delays for 15 seconds after sending the dial string to allow the phone connection to be established. The DIAL command should be followed by a WAIT15 CONNECT and an IF ERROR command:

```
INITMODEM
; dial the CCI BBS
DIAL ATDT1-415-873-6256
WAIT5
WAIT 15 CONNECT          ; wait for connect
IF ERROR QUIT            ; no connection
```

INITMODEM must be used before using DIAL to ensure that the modem is properly setup. The ATDT command before the number in the example is the Hayes Touchtone® Dialing command. Use ATDP before the phone number for pulse (or rotary) dialing.

The number of seconds to WAIT after a DIAL command for a CONNECT varies according to machine, modem, phone connection, etc. A number from 15 to 30 seconds is a good range.

DISCONNECT

Syntax: DISCONNECT

The DISCONNECT (or DISC) command hangs up the modem by dropping DTR. Use the DISCONNECT command immediately before the QUIT command to hang-up the modem before exiting the script.

FLUSH

Syntax: FLUSH

The FLUSH command discards all waiting characters in the communications channel input buffer. The FLUSH command can be used just prior to a SEND command to ensure all I/O buffers are clear and ready for the file transfer.

```
WAIT 30
FLUSH
SEND ERRORLOG.MED
```

IF

Syntax:
```
IF DOS command
IF CCI command
 IF ERROR command
IF EXISTS filespec command
```

The IF command allows conditional execution of script file commands. The IF DOS and IF CCI commands allow executing script file commands specific to DOS or CCI Concurrent DOS.

```
IF CCI CONSOLE ; use CONSOLE under CCI
IF DOS COM 3   ; use COM3 under DOS
```

The IF ERROR command allows special processing when a script command results in an error.

```
SEND MYFILE     ; try to send
IF ERROR DISC   ; send failed, disconnect
IF ERROR QUIT   ;   and exit
```

The IF EXISTS command checks if the specified filename exists. If the file is present the command is executed.

```
IF EXISTS MED.LOG CHAIN SENDMED
```

INITMODEM

Syntax: INITMODEM [String]

The INITMODEM (or INIT) command initializes the communications channel and sends an initialization string to the modem. If a string parameter is not supplied, INITMODEM sends the Hayes-compatible modem string:

```
AT &F Q1 &C1 &D2 &W
```

This wakes up the modem, reads factory configuration from ROM, disables result codes, turns DCD on only when modem detects carrier, hangs up the modem when DTR drops, and writes the changed configuration back to RAM. To ensure proper timing, always follow INIT or INITMODEM with a `WAIT 5` command.

LOG

Syntax:
```
LOG [filespec] [NEW]
LOG [filespec] [ON | APPEND]
LOG OFF
```

The LOG command sends TRANSIT status and error messages to a log file instead of displaying them on the console. If filespec is not specified, the default TRANSIT.LOG filename is used. The LOG NEW command erases any existing log file, creates a new log file, and begins logging. The LOG ON and LOG APPEND commands append log information to the end of an existing log file. If the log file does not exist, LOG ON and LOG APPEND creates a new log file. The LOG OFF command stops logging messages to the log file and starts displaying messages on the console again. Use LOG OFF with LOG ON or LOG APPEND to selectively log groups of commands in the script file.

If the NEW, APPEND, ON, or OFF options are missing, ON is assumed.

```
LOG GREGS.LOG NEW ; log to Greg's log
LOG OFF           ; turn off logging
```

OVERWRITE

Syntax: `OVERWRITE`

The OVERWRITE (or O) command allows the RECEIVE command to overwrite a file that already exists. If OVERWRITE is not used, RECEIVE generates an ERROR when trying to receive to a file that already exists. The default is not to allow RECEIVE to overwrite files.

```
OVERWRITE
RECEIVE AFILE.TXT
```

PDELAY

Syntax: `PDELAY` *#*

The PDELAY (or P) command specifies the number of system ticks TRANSIT waits between processing characters during a file transfer. A system tick is 1/60 of a second (about 16 milliseconds). The PDELAY command is followed by a number between 0 and 255 to specify the number of ticks to delay. The default PDELAY value is 1 tick. The PDELAY command is not valid under DOS.

Increase PDELAY if file transfers unacceptably slow down other tasks running under CCI Concurrent DOS. Increasing PDELAY slows down TRANSIT but increases performance of other programs.

```
PDELAY 2 ; set delay to 2 ticks
```

PUT

Syntax:
```
PUT [PCTERM | ASCII]
PUT String
```

The PUT command sets the PUT translation mode and sends a string through the communications port to the remote system. In PCTERM translation mode, PUT sends strings as IBM scan codes. PCTERM mode is necessary when TRANSIT is dialing into a terminal port defined as a PC terminal under CCI Concurrent DOS. In ASCII mode, PUT sends ASCII character strings. Use ASCII mode when communicating with a terminal port defined as a CRT (ASCII terminal) under CCI Concurrent DOS. The default translation mode is PCTERM.

```
PUT ASCII
PUT string sent as ASCII characters
PUT PCTERM
PUT string sent as scan codes
```

Use the PUT command to send account name and password strings to log into the remote system. Once at a command prompt on the remote system, use PUT to send commands to the remote system. When sending a string, PUT sends each character in the string and then sends a trailing carrige return. CTRL-A through CTRL-Z may be sent by preceding the character with a tilde (~). For example, a carrige return (CTRL-M) is sent by entering ~M.

PUT with no arguments sends a single carrige return. The special string PUT 2CR sends two carriage returns to the remote system. Use this when dialing into a CCI Concurrent DOS system to wake up the Concurrent Controls Security shell and get to the Username prompt.

```
; log into a CCI Concurrent PCTERM
; terminal port
PUT PCTERM
; send 2 carriage returns to wake up
; the Security shell then send an
; account name and password to login.
PUT 2CR
PUT JOE
PUT CDOSGURU
```

QUIT

Syntax: `QUIT [#]`

The QUIT (or Q) command exits the script file and terminates TRANSIT. Use the DISCONNECT command before quitting to ensure that the modem hangs up the phone.

On a QUIT, the DOS ERRORLEVEL is set to the current value of the ERROR flag. If QUIT is used with a numeric argument #, TRANSIT returns that value in ERRORLEVEL instead. The DOS ERRORLEVEL can be tested in a batch file when TRANSIT returns to DOS or CCI Concurrent DOS.

If the TRANSIT script file contained:

```
SEND MYFILE      ; send MYFILE
IF ERROR DISC    ; send failed, disconnect
QUIT             ; return ERROR value
```

the batch file could check the ERRORLEVEL returned by the SEND command:

```
TRANSIT @MYSCRIPT
IF ERRORLEVEL==41 ECHO send failed
IF ERRORLEVEL==0 ECHO send ok
```

Use QUIT # to define the error return values. For example, if the script file contained:

```
SEND myfile      ; send MYFILE
IF ERROR DISC    ; send failed, disconnect
IF ERROR QUIT 2  ; and return 2
QUIT 1           ; send ok, return 1
```

and the batch file contained:

```
TRANSIT @MYSCRIPT
IF ERRORLEVEL==2 ECHO send failed
IF ERRORLEVEL==1 ECHO send ok
```

the batch file takes the appropriate action as the TRANSIT SEND script command succeeds or fails.

RECEIVE

Syntax: `RECEIVE filename`

The RECEIVE (or REC or R) command receives a file through the communications port and saves it in the specified filename. If the file is received with no errors, the ERROR flag is set to 0. If an error occurred, ERROR is set to one of the TRANSIT RECEIVE errors.

```
RECEIVE myfile.txt ; receive file
IF ERROR DISC      ; it failed, disconnect
IF ERROR QUIT      ; and quit
```

RUN

Syntax: `RUN filename [Args]`

The RUN command executes an external .EXE, .COM or .CMD program file. When the external program exits, TRANSIT continues processing the script file.

```
RUN XCOPY FILE1 FILE2
```

When running a .CMD file, the .CMD file extension must be specified. When running a .BAT file, it must run through the CDOS.COM command shell. When running a .BAT file under DOS, use COMMAND.COM instead of CDOS.COM

```
RUN CMDFILE.CMD
RUN CDOS.COM /C BATFILE.BAT
```

SEND

Syntax: `SEND filename`

The SEND (or S) command sends the specified file through the communications port. If the SEND command is successful, the ERROR flag is set to 0. If an error occurred, ERROR is set to one of the TRANSIT SEND errors.

```
SEND MYFILE.TXT          ; send file
IF ERROR DISC            ; it failed, disconnect
IF ERROR QUIT            ; and quit
```

TIMEOUT

Syntax: `TIMEOUT #`

The TIMEOUT (or T) command sets the number of seconds TRANSIT waits while not receiving any characters from the remote system before assuming the remote system has lost contact (or timed out). The TIMEOUT parameter is a number between 0 and 255. The default TIMEOUT when sending a file is 60 seconds. The default TIMEOUT when receiving a file is 30 seconds.

```
TIMEOUT 120 ; set timeout to 2 minutes
```

WAIT

Syntax:
```
WAIT #
WAIT [# | ?] [CONNECT | "string"]
```

The WAIT # command delays a specified number of seconds before processing the next script command. Use WAIT # to delay an absolute number of seconds to allow the remote system to do something. For example, if using PUT to tell a remote system to perform a file copy, to WAIT a few seconds allows the remote system to complete the copy before PUTing the next command.

```
PUT XCOPY BIGFILE BIG2  ; do big copy
WAIT 10                 ; wait a bit
```

The second form of the WAIT command allows delaying until a specified condition occurs. The WAIT # CONNECT command waits to establish a modem connection. If the number of seconds expire and the modem has not connected (that is, the modem carrier signal is not detected), the ERROR flag sets. If the modem is connected by the time the number of seconds elapse, the ERROR flag is 0. Use WAIT # CONNECT after a DIAL command to ensure that there is a successful connect to the remote modem:

```
DIAL ATDT8736256; dial CCI BBS
WAIT 10 CONNECT ; wait for carrier
IF ERROR QUIT   ; we didn't connect?
```

The WAIT # string command delays a number of seconds, waiting until the specified string is received from the remote system. The string must be enclosed in quotes (") and must not be more than four characters long.

Both the WAIT CONNECT and WAIT string command can be told to delay for an absolute number of seconds (#) or can be told to delay forever using a question mark (?). The question mark is not valid for a simple WAIT command.

For example, the following script instructs the local CCI Concurrent DOS machine to wait forever until another machine calls in and sends a special string before beginning a file transfer. This could establish a form of security on the local machine's dial-in port since it is very unlikely someone would dial in and send the exact string.

```
AUX 0           ; select COM1 under CCI
WAIT ? "HEYU"   ; wait for "HEYU" string
RECEIVE AFILE   ; start transfer
```

XMODEM

Syntax: XMODEM

The XMODEM (or X) command tells TRANSIT to use the XMODEM file transfer protocol. XMODEM transmits files in 128-character blocks. When TRANSIT begins a file transfer with the remote file transfer program, TRANSIT first tries the XMODEM-CRC protocol before defaulting to the XMODEM-CHECKSUM protocol. The XMODEM-CRC protocol can be more effective at detecting transmission errors.

YMODEM

Syntax: YMODEM

The YMODEM (or Y) command tells TRANSIT to use the YMODEM file transfer protocol. YMODEM transmits files in 1,024 character blocks and is generally more efficient than XMODEM protocol. On a noisy phone line, when line noise corrupts data in a block, TRANSIT is required to retransmit

the entire block. In this case, XMODEM may achieve better throughput than YMODEM because it retransmits smaller blocks.

%N

Script file replacement variables: %1 through %9. Use replacement variables to pass parameters from the TRANSIT command line to commands in a script file. For example, if the script file contained:

```
AUX %1    ; select aux port
%2        ; select protocol
SEND %3   ; send file
```

and TRANSIT executes with the command line:

```
TRANSIT @SCRIPT 3 XMODEM REPORT
```

TRANSIT replaces the %N variables in the script file with the parameters from the command line and executes:

```
AUX 3          ; select aux port
XMODEM         ; select protocol
SEND REPORT    ; send file
```

Replacement variables are also useful for using login passwords in script files. If a password is placed in a script file anybody can TYPE the script file and see the password. If a replacement variable is used in the script file instead, for example:

```
PUT JSMITH   ; send account name
PUT %1       ; send password
```

and the login password is supplied as a command line parameter when executing the script file:

```
TRANSIT @MYSCRIPT XYZZY
```

TRANSIT replaces the %1 variable in the script file with the command line parameter and sends XYZZY to the remote system as the login password.

; Comments

TRANSIT treats text after a semicolon (;) as a comment. Comments can appear on their own line or appear after a command, as long as there is a space or tab between the command and the semicolon. Use comments to add descriptions to the script file.

```
;this is a comment
QUIT ;this is a comment too
```

TRANSIT Script Examples

This section contains several example script files. Invoke scripts with the command:

```
TRANSIT @SCRIPT [User] [Pw] [Protocol] [FName]
```

where:

User	is the login account name
Pw	is the login account password
Protocol	is XMODEM or YMODEM
FName	is the name of a file to download

The SCRIPT file contains:

```
; SCRIPT - a sample TRANSIT script to dial
; into a PCTERM terminal port on a CCI
; Concurrent system and download a file.
; The script command line parameters are:
;   %1 = user name
;   %2 = user password
;   %3 = protocol to use (xmodem or ymodem)
;   %4 = filename to download (receive)
IF CCI AUX 6            ; under CCI, use addon port 5
IF DOS COM 2            ; under DOS, use COM2
LOG NEW                 ; log to TRANSIT.LOG
INITMODEM               ; initialize modem & dial
DIAL ATDT1-415-873-6256
WAIT 10 CONNECT         ; wait for connect
IF ERROR QUIT           ; if no connect, quit
; at this point we are connected.  we
; now log into the CCI system
PUT PCTERM              ; send scan codes
PUT 2CR                 ; send <return><return> to
                        ; get to "Username" prompt
PUT %1                  ; send name<return>
PUT %2                  ; send password<return>
WAIT 15 "C:\"           ; wait for command prompt
IF ERROR DISC           ; didn't get to prompt?
IF ERROR QUIT           ;   disconnect and quit
; tell the remote system to send us file
; %4 using the desired protocol in %3.
PUT TRANSIT %3 SEND %4
; the remote system is now waiting for us to
; start the file transfer.  setup our end.
%3                      ; use same protocol
OVERWRITE               ; allow overwriting file
RECEIVE %4              ; and receive the file
IF ERROR DISC           ; receive failed?
IF ERROR QUIT           ; disconnect and quit
PUT LOGOUT              ; send "LOGOUT" to remote system
LOG OFF                 ; stop logging
DISCONNECT              ; shut down modem
QUIT                    ; leave
```

Use the TRANSIT /V option to test the script before running it:

```
TRANSIT @SCRIPT /V MARIA MPASS YMODEM MEDMGR.LOG
```

■ *Example 2 :*

Here are two script files that transfer information from one location to another location for updating of data files.

A company had two locations that were tracking inventory and wanted to periodically update the warehouse files from their front store. The transfer of information needed to be automatic, with no operator intervention necessary.

The system is set up in the warehouse with COM1 defined as AUX and a modem connected to it; the system in the front store has COM1 defined as PCTERM. This was necessary since transit scripts RUN on consoles and FEED aux ports. A terminal port can not run a transit script that uses its own port as the transfer mechanism.

Two scripts were used; one sends a file to the store, the other gets a file from the store. Both execute on any console of the warehouse computer.

The warehouse computer is configured with security enabled — a mandatory to achieve reliability on script execution. LOOKOUT is monitoring COM1, which is a REMOTE console, and the disconnect for bad-names has been increased to 5 from 3. A simple name and password were used.

The protocols for the store would be RTS - CTS - XPC - DSR - DCD. The warehouse is RTS - CTS only. The modems are set for RTS-CTS only, in addition to normal factory defaults. They also disconnect on DTR drop (usually AT&D3) and provide a true DCD (AT&C1). To reduce the amount of noise produced, the modem is configured to not echo commands or result codes.

```
;SCRIPT to send file to store from warehouse
;log new                                    ;log if needed
aux 0                                       ;assign port
xmodem                                      ;simple protocol
PUT PCTERM                                  ;take scan codes
INITMODEM                                   ;reset modem
DIAL ATDT5551212                            ;call store
wait 10 connect                             ;extra time for smart modems
if error quit 1                             ;no connect -exit
run force aux0 [1D][38][44][C4][B8][9D]~    ;clear all keys
wait 5                                      ;settle time
flush                                       ;clear aux buffer
WAIT 1                                      ;settle
PUT 2CR                                     ;wake up tmp
WAIT 1                                      ;response time
PUT JOE                                     ;name
wait 1                                      ;settle
PUT CCI                                     ;password
wait 5                                      ;wait for batches to complete
PUT TRANSIT R TESTFILE.LZH 0                ;send command to receive
WAIT 1                                      ;settle
FLUSH                                       ;clear buffer
SEND TESTFILE.LZH                           ;ship it
if error disconnect                         ;fail? restart
PUT LOGOUT                                  ;happy, send logout
;log off                                    ;stop log file
disconnect                                  ;drop dtr
RUN CCIMODE COM1 9600 RESET                 ;reset port
quit                                        ;all done
```

```
;SCRIPT to get file from store
;log new                                  ;if desired
aux 0                                     ;select port
xmodem                                    ;select protocol
PUT PCTERM                                ;use scan codes
INITMODEM                                 ;prep modem
DIAL ATDT5551212                          ;call number
wait 10 connect                           ;wait for handshaking
if error quit 1                           ;exit if fail
flush                                     ;clear buffer
run force auxO [1D][38][44][C4][B8][9D]-  ;clear noise
WAIT 1                                    ;settle
PUT 2CR                                   ;wake up tmp
WAIT 1                                    ;settle
PUT JOE                                   ;send name
wait 1                                    ;settle
PUT CCI                                   ;send password
wait 5                                    ;wait for batches
FLUSH                                     ;clear buffer
PUT TRANSIT S TESTFILE.LZH                ;send the send command
WAIT 1                                    ;settle
OVERWRITE                                 ;wipe old file
RECEIVE TESTFILE.LZH                      ;get the data
if error disconnect                       ;bad news - retry
PUT LOGOUT                                ;say bye
;log off                                  ;close file
disconnect                                ;drop dtr
RUN CCIMODE COM1 9600 RESET               ;clear port
quit                                      ;done
```

TRANSIT Script ERROR Codes

This table lists the TRANSIT error messages and their ERROR numbers. When exiting a TRANSIT script, the QUIT command sets the DOS ERRORLEVEL to the current ERROR number. The QUIT # command overrides ERROR with a user defined error number.

In some instances, an error code applies to more than one message. This provides for simpler error testing in batch files where a group of errors result in the same ERRORLEVEL.

Success code

00	File transfer completed successfully or TRANSIT termination without errors.

Command line or script file parameter errors

10	Improper number of command line parameters.
11	Specified AUX or COM port [#] is out of range. If using an AUX or COM port that is invalid for the system, try using another AUX or COM port. Use PRINTMAP /A for a list of valid AUX port numbers under CCI Concurrent DOS.

12	First command line parameter must be COMx on a DOS system. If running TRANSIT from the command line under DOS, specify a COM port to use.
13	First specify S (or Send) or R (or Receive) before specifying a file name. Parameter being parsed is [...]. When running TRANSIT from the command line, use the SEND or RECEIVE command before supplying a filename. This error can also occur if the "@" character is not included before a TRANSIT script file name.
14	Invalid command [...]. If OVERWRITE parameter is intended here, correct syntax is O or OVERWRITE.
15	Invalid numeric format for TIMEOUT [...] or parameter greater than maximum value of 255.
16	Invalid command [...]. If TIMEOUT parameter is intended here, correct syntax is T xxx or TIMEOUT xxx.
17	Invalid numeric format for PDELAY [...] or parameter greater than maximum value of 255.
18	Invalid command [...]. If PDELAY parameter is intended here, correct syntax is P xxx or PDELAY xxx.
19	Invalid numeric format for WAIT [...].
25	Invalid script argument [...].

TRANSIT validation errors

30	For file transfers through the console, TRANSIT must be executed from a FOREGROUND console only.
31	Concurrent Controls header mismatch.
32	CCPM.SYS Revision level mismatch. This version of TRANSIT is not compatible with the running version of CCI Concurrent DOS.
33	Non-serial console type. Only transfer on serial ports. For host or MaxStation consoles, use the TRANSIT AUX option.
34	Specified AUX channel [#] is same as the console channel [#]. It must be different.
35	Specified COM port [#] does not exist on this system or is defined as a terminal port and can not be used.

	Specified AUX port [#] does not exist on this system or is defined as a terminal port and can not be used.
	Cannot set AUX port [#].
36	Specified COM port [#] is in use by another console.
	Specified AUX port [#] is in use by another console.
	Cannot attach to AUX port [#].
37	No port has been specified. Before executing the script command that caused this error, a communications port must be specified to use with a CONSOLE, AUX, or COM script command.
38	Modem has not been initialized. Before executing a DIAL command the modem must be initialized with an INITMODEM script command.
39	Cannot receive file, initialization is not complete.
	Cannot send file, initialization is not complete.
	Cannot transfer file, initialization is not completed.

TRANSIT file I/O errors

Errors when SENDing files:

41	Cannot access specified file [FN]. It may be read- only or a directory.
	Cannot open specified file [FN]. Too many open files.
	Specified file [FN] does not exist.
	Problem opening specified file [FN].
42	Bad file handle — file may have been erased during transfer by another task.

Errors when RECEIVEing files:

46	Cannot create temporary file [temp.$$$]. Too many open files.
	Problem creating temporary file [temp.$$$].
47	Cannot access specified file [FN]. It may be read- only or a directory.
	Cannot create specified file [FN]. Too many open files.

Problem creating specified file [FN].

48 Cannot rename temporary file [temp.$$$]. Transferred file is named TEMP.$$$.

Problem renaming temporary file [temp.$$$]. Transferred file is named TEMP.$$$.

49 File already exists. Tried to RECEIVE to a filename that already exists on disk. Use the OVERWRITE command before the RECEIVE to overwrite files.

Errors when checking IF files EXISTS:

55 Cannot access specified file [FN]. It may be read- only or a directory.

Cannot open specified file [FN]. Too many open files.

Specified file [FN] does not exist.

Problem opening specified file [FN].

Script command errors

RUN errors:

61 Argument list or environment data too large.

Run file [FN] does not exist.

Run file [FN] cannot be executed.

Not enough memory to execute run file [FN].

Problem processing run file [FN].

WAIT errors:

65 Timeout [# seconds] occurred before CARRIER detected.

66 Timeout [# seconds] occurred before STRING detected.

Log file errors

81 Cannot access TRANSIT.LOG file. It may be read-only.

Cannot create TRANSIT.LOG file. Too many open files.

Cannot open TRANSIT.LOG file. Too many open files.

Problem creating TRANSIT.LOG file.

Problem opening TRANSIT.LOG file.

82 Cannot access TRANSIT.LOG file. Disk may be full.

Abort conditions

91	Transfer ABORTED by user.
92	Transfer ABORTED by remote computer.
93	Transfer ABORTED — No response from remote computer. Connection with remote computer was never established or was lost.
94	Transfer ABORTED — Too many errors from remote computer. Try a lower baud rate or better phone line.
95	Transfer ABORTED — Lost sync with remote computer. At least one data packet was lost. Try a lower baud rate or better phone line.

Chapter 6:
Scheduler System

Scheduler System

CCI Concurrent DOS includes a system for job scheduling and automatic execution. It allows the execution of a program or batch file at a specific time and date. The scheduler is useful for running tasks on a repeating basis (such as running a tape backup every night) or for deferring a task to a later time (such as doing a file transfer late at night).

The scheduling system consists of two programs — SCHEDULE.EXE and JOBMGR. COM. SCHEDULE specifies the time(s) and date(s) programs and batch files to be executed. All scheduled tasks are stored in memory. JOBMGR, the task or job manager, runs in the background and executes all scheduled tasks at their appointed time.

If using security, use the DO-UPON-BOOT feature of the Concurrent Controls Security system to start the JOBMGR program as part of the CCIBOOT.BAT file. Since the list of scheduled jobs is stored in RAM memory while the operating system runs, recurring jobs such as tape back-ups have to be rescheduled when restarting the system.

If not using security, add an entry to one of the STARTXXY.BAT files; for example, START004.BAT, to invoke JOBMGR. JOBMGR should only be invoked on a single session.

Scheduling Jobs (SCHEDULE)

A task is scheduled by running SCHEDULE at the command line or in a batch file. A task is scheduled to run at a certain time on a certain day, up to one year in advance. The day is specified by the day of the week or numerically as a day of the month or year. The time is specified as an absolute time in hours and minutes or as an offset from the current time. The basic syntax to schedule a job is:

```
SCHEDULE program ON user# day time
```

For example:

```
SCHEDULE BACKUP.EXE ON 5 MON 1430
```

runs the program BACKUP.EXE on user number 5 (usually, the first session on the first terminal) on Monday at 2:30 PM. The complete SCHEDULE command line syntax is:

NOTE: *Scheduler does not execute on a session if another program is already running at the time that the scheduled job is to run.*

```
SCHEDULE [/Help] [command | "command"]  [ON # | ON @ | ON #-#]
  [day] [time] [EVERY] [EVERY DAY] [WAIT n] [RETRIES n]
  [NOWAIT] [/Reset] [/Delete] [/Status]  [/Unlock]
```

Where:

/Help displays a help screen

command The program or batch file to execute. This parameter must come first. The command can be up to 24 characters long. If a longer command line is required, put the command line in a batch file and schedule the batch file.

"command" Enclose the command in double quotes (") if the scheduled command line contains spaces.

ON Use ON # to specify the virtual console that the job runs on. The host console begins with user number 1, the first terminal usually begins with user number 5, etc. For example, ON 4 means "run the job on user number 4, the fourth session on the host console".

Use ON @ to have Job Manager run the job on the current console (that is, the console the job was SCHEDULEed from).

If a job is already running on a virtual console at the scheduled time, that console cannot be used to execute the scheduled command.

Use ON #-# to specify a range of user consoles, as in ON 1-4. If the first virtual console is not available, the Job Manager tries to run the job on the next virtual console in the range. This continues until the Job Manager tries the last console in the range. Use ON #-@ or ON @-# to specify a range ending or beginning with the current console.

Use a question mark (?) as a wildcard for the last console available on the system. For example, ON 1-?, allows the Job Manager to try all consoles on the system. Use ON @-? to specify a range beginning with the current console and ending with the last console.

day The day to schedule the job. Possible values are: MON, TUES, WED, THURS, FRI, SAT, SUN, TODAY and TODAY+ XX. The XX following TODAY+ is the number of days in the future to schedule the job. The number of days can range from 0 to 366.

time NOW NOW+	The time to schedule the job. Possible values are: HHMM and NOW or NOW+ XX. HHMM is the hour and minute to execute the job in a 24-hour clock format. NOW means to run the job immediately. NOW+ XX indicates to execute the job XX minutes from now. The number of minutes can range from 0 to 1440 (1 day). When NOW or NOW+ is used the day and time should not be specified. The NOW and NOW+ options are a good way to test a batch file, ensuring that the scheduled job does work.
EVERY XXX	An optional parameter to schedule a job every week at the same day and time. XXX is the day of the week (MON, TUE, WED, THU, FRI, SAT, or SUN) to run the job.
EVERY DAY	An optional parameter to schedule a job every day at the same time.
WAIT	An optional parameter that specifies the number of minutes the Job Manager waits before trying to reschedule the job because the specified console(s) is/are unavailable. Use WAIT with the RETRIES parameter. If WAIT is not specified, the Job Manager waits one minute.
RETRIES	Use RETRIES # to specify the number of times the Job Manager can reschedule the job before giving up. If the number of retries expire and the job still cannot be run, the job is deleted. If RETRIES is not specified, the Job Manager only retries once. Use RETRIES ? to have the Job Manager try to schedule the job forever.
NOWAIT	Use NOWAIT to force the Job Manager to try to start the job even if the console is unavailable.
/Reset	Deletes all scheduled jobs. Confirmation to delete the currently scheduled jobs is asked.
/Delete	Deletes a single scheduled job from the joblist.
/Status	Displays a list of all scheduled jobs.
/Unlock	This option is provided in case an unforeseen error occurs and the SCHEDULE program returns a `Semaphore locked` error message. Confirmation to unlock the scheduler is asked.

- *Example 1: Schedule the tape backup program (or batch file) TBACKUP to execute on user console 4 every Sunday at 11:30 PM.*

```
SCHEDULE TBACKUP ON 4 EVERY SUN 2330
```

- *Example 2: Schedule the program (or batch file) REMINDER to execute on user console 1 in 30 minutes.*

```
SCHEDULE REMINDER NOW+ 30 ON 1
```

Use NOW to execute REMINDER.BAT immediately as a test of the batch file's code.

- *Example 3: Schedule a backup command line including command line parameters, to execute every Sunday at 11:30 PM. Try to run the program on any available console except the host consoles. (1-4)*

```
SCHEDULE "BACKUP C:" ON 5-? EVERY SUN 2330
```

- *Example 4: Schedule a TRANSIT script to execute a file transfer every Thursday night at 10 pm. If the TRANSIT fails, log the time and date of the error and use the NOW+ option to reschedule the script every 15 minutes until the TRANSIT succeeds.*

```
; schedule senderrs.bat every thursday at 10pm
SCHEDULE SENDERRS.BAT ON 4 EVERY THURS 2200

; senderrs.bat - a batch file to run the transit
; script sendfile.sf. if the script fails, log the
; error and reschedule for 15 minutes from now.
DATE /B /E
TIME /B /E
; log start date/time, initiate transfer
ECHO >>SENDFILE.LOG Transit on %DATE% AT %TIME%
TRANSIT @SENDFILE.SF
IF ERRORLEVEL==0 GOTO SUCCESS
; transit failed, log and reschedule
ECHO >>SENDFILE.LOG Failed, error %ERRORLVL%
SCHEDULE SENDERRS.BAT NOW+ 15 ON 4
GOTO DONE
:SUCCESS
; transit ok, log elapsed time
TIME /B /E
ECHO >>SENDFILE.LOG Elapsed time %ELAPSED% seconds.
:DONE
```

Executing Jobs (JOBMGR)

The Job Manager program runs in the background and is periodically asked to run a scheduled task. The Job Manager tries to run the task on the console specified on the Scheduler command line. If the console is unavailable, the Job Manager may try to reschedule the task. All other scheduled

tasks wait until the Job Manager either successfully runs the current job, or the number of RETRIES expire and the Job Manager goes onto the next job.

The JOBMGR command line syntax is:

```
JOBMGR [/H] [/S]
```

Where:

/H	displays a help screen
/S	terminates a currently running JOBMGR

JOBMGR with no parameters starts the Job Manager. JOBMGR should be listed in the STARTxxy.BAT or CCIBOOT.BAT file so that it is started when the system is booted.

This page intentionally left blank.

Chapter 7:
Menuing System

Menuing System

Menus provide a user-friendly interface that can be combined with the Security System to limit which programs and functions a user can execute. The Menuing System develops menus that provide users with choices, such as executable programs, utilities, or even batch files. A choice is automatically invoked when selected by the user. Menus can prevent a user from ever seeing a system prompt.

Files, Structures, and Requirements

The parts that comprise the Menuing System are as follows:

EDITMENU Creates and edits the menu screens.

RUNMENU Executes the menu screens.

COPYMENU Duplicates the menus and reduces the size of the data files after multiple edits.

TERM.DAT These files contain translation codes that display the menu screens on different types of terminals. TERMCOLO.DAT is used for color terminals; TERMMONO.DAT is used for monochrome terminals; and TERMTTY.DAT is used for ASCII terminals.

MENU.DAT The menu data file contains the screens that are presented to the user. MENU.DAT is the default filename and can be changed.

The Menuing System is invoked with the command RUNMENU. RUNMENU assumes that TERM.DAT is in the same subdirectory as RUNMENU, that the first screen is called MAIN, and MENU.DAT is in the currently active subdirectory. If the menu data file names are different, the complete syntax is:

```
RUNMENU <mymenu.dat> <screen>
```

The menu name is supplied for <mymenu.dat>, and the screen name is supplied for <screen>. Should there be a menu file called EDITING.DAT, and a screen called WORDPR, then the command is:

```
RUNMENU EDITING.DAT WORDPR
```

The terminal file, TERM.DAT, must be in the same subdirectory as RUNMENU. RUNMENU can be in more than one directory and each directory can have a specific TERM.DAT. Add the console number to TERM, as in TERM1, TERM4, etc., to make specific translation files. Since the menu screens are also specific for various users, this is usually not needed.

When a task is executed from within RUNMENU, the user is automatically returned to the original directory. RUNMENU cannot be used to change directories. It is not possible to exit RUNMENU and be located in a different directory. RUNMENU is designed to provide a method of keeping users away from the system prompt; it is not a navigation program.

The memory needed for each copy of RUNMENU is 16k. Another 16k is required at the time each menu is displayed.

Terminal Data Files

The supplied TERM.DAT files are as follows:

TERMMONO.DAT	This is for monochrome workstations.
TERMCOLO.DAT	This is for color workstations.
TERMTVI.DAT	This is for ASCII terminals in TVI 920 Emulation.

RUNMENU only looks for a TERM.DAT or a TERMx.DAT. The x is a digit from 1 to 99. This x is the same as the USER number seen as part of the command line prompt. When mixing color and monochrome workstations, it is best to add user numbers to the TERM.DAT files. Initially, the use of a single monochrome TERM.DAT is recommended. Color menu screens on a mono terminal are ignored. Mono menu screens on a color terminal displays in black and white.

TERMTTY.DAT	A simple file with no special features.
TERM.DAT	A copy of TERMMONO.DAT.

Built-in Commands

There are only three built-in commands:

- PRINTER
- CHDIR
- CD

Any other command causes RUNMENU to invoke SHELL.COM for execution. The SHELL program is equivalent to the DOS COMMAND.COM, and is usually found in the CDOS subdirectory. When SHELL is used, a brief message shows that the SHELL command is being executed. Also, the user's status line usually disappears when SHELL is loaded, as with any DOS program.

SHELL is also loaded when a batch file executes. If executing a task directly, a small amount of memory is saved.

The printer assignment can be changed from within RUNMENU and this assignment remains until RUNMENU is exited, but the original printer assignment resumes when RUNMENU exits.

The directory may change while within RUNMENU, but when the menuing system is exited, the user is returned to the original subdirectory.

NOTE: *To change drives, use CD D: as a built-in command.*

When external commands are used

The command shell CDOS.COM is loaded to execute any external command. The built-in commands are CD, CHDIR, and PRINTER. CDOS.COM loads to run a batch file. If possible, execute commands directly.

Building Menus — EDITMENU

While EDITMENU functions on terminals, the following makes a menu on the main console.

There are two sample menus supplied: PREPMENU.DAT which is the menu used during an installation, and HDMENU.DAT.

■ *Example: Create a new menu.*

First, log into the CDOS subdirectory. Check that there is not an existing MENU.DAT by doing a DIR of MENU.DAT:

```
C:\CDOS>DIR MENU.DAT
```

It should return `FILE NOT FOUND`. If one exists, change its name.

To create a new MENU.DAT enter the command:

```
EDITMENU
```

The message that MENU.DAT is not found or is read-only should display. Hit the F2 key to create a MENU.DAT

The screen displays several options. The F2 key allows the creation of a new menu. Hit F2, and enter `MAIN`.

A screen appears with a list of command options. Enter:

```
FIRST MENU
```

The next operation is to hit F5 to center the line. After F5, hit the ENTER key several times to move lower on the screen. Then enter:

```
F1 = TREE
```

This will be the only command for this menu. Hit F10 to exit.

A screen now displays that permits the entry of a command for the function keys. Under the TYPE field, enter A, since this is a command to

execute. The cursor now moves to the entry field, where TREE is entered. Hit F10 to exit. The menu is saved with an ENTER key. Then the main entry menu displays. Exit EDITMENU with the Escape key.

To test the menu, type RUNMENU at the prompt. The first screen should appear; hitting F1 should invoke the TREE command.

EDITMENU Options

Opening menu:

```
                    MENU CREATE AND EDIT (11/25/91)



Please Select:

          F2  Make New Menu
          F3  Change Existing Menu
          F4  Copy Existing Menu
          F5  Delete Menu
          Esc Exit Editmenu
```

While editing screens:

Ctrl Z	Move Commands. This moves the help screen of a command to the upper or lower part of the screen, allowing all 24 lines to be displayed.
ESC	Exit Without Changes. Abort the edit session.
F1	HELP. Explain the options available in this edit session.
F2	Spec. Effects. Menu driven method of adding features.
F3	Insert Line. Add a blank line.
F4	Delete line. Remove the line where the cursor is located.
F5	Center Line. Center the text on the cursor line. (Center the text before adding borders or special effects.)
F6	Copy Lines. Allows duplication of lines.
F7	Cursor Position. Displays current cursor position.
F8	Display Screen. Show screen as it should appear. This may not always be accurate, depending on the target workstations' TERM.DAT abilities.

F9	Special Chars. Select special graphics characters. The actual menu display may not be accurate, depending on the TERM.DAT used.
F10	Done. Finished with menu, go to command screen.

Command Screen:

```
                    MENU CREATE AND EDIT (11/25/91)
Fcn    Type   Menu Name or Command Text
 1

 2

 3

 4

 5

 6

 7

 8

 9

10

Press ENTER or TAB for next field, Shift TAB for prior field
      F1(HELP), F2(Disable ESC Option), F10(Done)          Esc(Prior Screen)
```

FCN	Function key number.
Type	A is a command to execute. M is a menu to display. E is exit to the command line. A blank disables the key in this menu.
Menu Name or Command Text	This is where to put the command(s) to run, 128 characters maximum.

NOTE: *Multiple commands can be separated with an exclamation point (!). Do not use leading spaces before the exclamation point.*

Command Options display:

```
Disable ESC to return to prior menu level? (Y/N) N
   NOTE:  The prior menu level is the system
   prompt if this is the main menu.
```

This disables the ESCAPE key's ability to backtrack one menu level or to exit to the system prompt.

```
Use 1,2,3...instead of F1, F2,... (Y/N) N
```

This causes the menu to accept numbers instead of function keys. Only valid when used with PCTERM-type workstations; ASCII workstations always use numbers.

```
SKIP the [ANY KEY] wait after commands? (Y/N) N
```

This suppresses the message to hit any key when a command finishes executing and returns the user directly to the menu screen.

```
Command to execute at start of menu:
```

Enter a command for RUNMENU to execute when this screen is invoked.

```
Command to execute at end of menu:
```

Enter a command to execute when RUNMENU is leaving this menu.

Compressing a Menu — COPYMENU

A menu file gets quite large after several edits. Use the following procedure to compress the menu file:

1. If the menu is called MENU.DAT, change its name to TEMP.DAT. Ensure that there is no MENU.DAT in the working subdirectory.
2. Create a MENU.DAT by executing EDITMENU, then hit F2 to create a MENU.DAT, then hit Escape.
3. Invoke COPYMENU.
4. Enter TEMP.DAT as the source (or the menu name).
5. Enter MENU.DAT as the destination name.
6. Hit F10 to begin.
7. Enter an X next to each screen of the data file.
8. Enter F10 to begin the process.
9. Copy MENU.DAT to the file name.

10 Test the new menu. Continue on to the section on ERROR MESSAGES if errors occur.

Error Messages

▼ `Can't Open Menu File (not in current subdirectory)`

Indicates that the menu data file cannot be found. RUNMENU uses MENU.DAT as the default. If a different name is used, the entire name is required. The menu data file must be in the current directory; directories in the PATH are not searched.

▼ `Can't Read Menu (repack and/or check name)`

The default name for the first menu screen is MAIN. If starting with a different screen name, that name must be supplied with the data file name when RUNMENU is invoked.

If the menu data file is very large it may need cleaning up with COPYMENU. This procedure is described earlier. This procedure is only needed after several edit sessions. It is not needed in normal use.

▼ `Specified Menu Screen Not Found`

Check the spelling that was typed.

▼ `Can't Allocate (16k more banked) Memory`

During the screen display process RUNMENU needs a 16k block of memory to read the data file. If there is no memory left, or the system is running a non-banked program (like some communication tasks), then menus cannot be changed.

▼ `Menu Corrupted, repack with EDITMENU/COPYMENU`

When the menu data file gets over 64k, the internal buffering may not be enough to find a menu. See the previous re-pack procedure.

▼ `RUNMENU cannot find a TERM.DAT for this terminal`

RUNMENU looks for a TERM.DAT in the directory where it was loaded. A TERM.DAT must be there also. If using several directories, the put a RUNMENU and TERM.DAT in each one.

If RUNMENU is in the CDOS directory, then TERM.DAT must be there also.

If RUNMENU is in a special batch file directory, then TERM.DAT must be in that special directory also.

Menu data files must be in the current directory. They are never searched for as part of the path.

▼ `Bad command or filename from execute attempt`

After RUNMENU tries a command internally, it looks for the command on the disk. If the command is found and it is not a batch file, RUNMENU tries to execute it. If it cannot be executed, this message displays. There may be insufficient memory. If the command does not exist at all, then RUNMENU asks SHELL to run it. If SHELL fails, SHELL supplies it's own messages.

▼ `Requires CCI Concurrent version 3 or 6`

This version of RUNMENU needs CCI Concurrent DOS 6.21 or 3.01 or higher.

▼ `Invalid directory requested`

A CD command to a directory that does not exist was attempted. The full directory name must be used.

▼ `Physical Error accessing disk`

This is a serious problem that needs correcting. Use `CHKDSK /M` to verify the integrity of the drive.

▼ `Invalid printer number for this computer, IGNORED`

Only valid printer numbers are permitted. A non-existent printer cannot be assigned.

Chapter 8:
Print Spooler System

Print Spooler System

The CSPOOL print spooler allows multiple users to share printers under CCI Concurrent DOS. When using a multiuser, multitasking operating system, more than one user may try to use the same printer at the same time. Users contend for printer resources and if there is no spooler, the first user to start printing attaches to the printer. This blocks other users from using the printer until the first user releases (or detaches from) the printer. A program releases a printer by either using an operating system service to instruct CCI Concurrent DOS to detach or by exiting, which causes CCI Concurrent DOS to automatically detach the program from the printer.

The spooler solves the printer contention problem by intercepting print output from an application and rerouting it to a "spool file" that is created on the hard disk. If multiple programs try to print to the same printer, the spooler creates individual spool files for each program. When a program finishes printing and releases the printer, that spool file is closed and sent to the printer. The entire process is called a print job, or simply, a job. The spooler gives each print job a unique job number by which it can be referenced. The print job that the spooler is currently sending to a printer is called an active print job. A print job waiting for an available printer is called a queued print job.

Many DOS-based programs do not explicitly tell the operating system to release the printer. When the spooler is not running, these programs stay attached to the printer until the user exits the program, preventing all other applications from accessing that printer. Also, programs that do not release the printer need some way to instruct the spooler to begin printing their output, since no "detach" occurs. The spooler resolves this by automatically detaching a program from a printer if the program stops sending print output for a few seconds. The time between when a program stops sending characters and the time when the spooler automatically detaches the program from the printer is called the timeout period.

Besides eliminating printer contention between multiple users and automatically detaching DOS-based applications from the printer, there are other benefits to using the spooler. For example, since programs are essentially printing to a spool file on a high-speed hard disk rather than a slower printer, control of the program is quickly returned to the user. This allows the user to return to working as the spool file independently sends data to the printer. Thus, while the total elapsed time from starting a print job to having a hard copy may be longer when using the spooler, overall work productivity is increased.

Another benefit the spooler provides is the automatic printing of a formfeed after a print job to ensure each job begins at the top of a new page. This is for applications that do not advance the paper when a job is finished. The spooler can also assign different job priorities to print jobs so

the output from time-critical applications is printed before output from other programs. The spooler can also map one printer onto another printer. This means the spooler can send print output meant for one printer to a different printer instead.

There are some disadvantages to using the print spooler. Depending upon the application and print jobs, the resulting spool files can be very large. It may require a few megabytes of free hard disk space to hold the spool files. The spooler requires a minimum of one megabyte of free disk space to begin loading.

The spooler can only send print jobs to four different physical printers simultaneously. If there are more printers, and the spooler is currently printing to four of them, print jobs for the other printers wait until another print job is finished before starting to print.

As mentioned above, the spooler does not physically print until the application has completed its print output and released the printer. This means the elapsed time from starting the print job to when the printer is finished printing can be longer when printing through the spooler. If the spooler has to wait for an application to timeout, a longer delay can occur.

One advanced feature of CCI Concurrent DOS CSPOOL is that the print spooler can also spool CTRL-P and PrintScreen output. If CTRL-P or PrintScreen is pressed, and the default printer is a spooled printer, the print output is spooled. However, the spooler may timeout a CTRL-P print job if the screen display is inactive for the timeout period. The print spooler also intercepts output redirection to the printer. For example, spool a file by copying or typing it to the PRN device:

```
COPY FILE PRN
TYPE FILE >PRN
```

The following sections discuss how to use the CSPOOL print spooler program; how to configure applications for use with the spooler; how to install the spooler; and how to use the CONFIG.CPL spooler configuration file to customize the spooler for the environment.

CSPOOL Commands

The CSPOOL program starts and stops the print spooler and is also used while the spooler is running to display, change, or delete currently active and queued print jobs. The command line parameters used with the CSPOOL program are:

BOOT	Alternative for starting the spooler.
START	Start the spooler.
QUIT	Stop the spooler.

STATUS	Display a status report.
STOP	Delete a job.
PRIORITY	Change a job's priority.
NOSPOOL	Make a printer non-spoolable.
SPOOL	Make a printer spoolable.
SUSPEND	Temporarily stops spooling to a printer.
CONTINUE	Continues spooling to a suspended printer, continuing any active print job.
RESTART	Continues spooling to a suspended printer, restarting any active print job from the beginning of the spool file.
OWN	Reserve a printer for a single console.
RELEASE	Release an owned printer.

The following section describes each command.

BOOT

Syntax:
```
CDSPL BOOT
ESPL BOOT
CSPOOL BOOT
```

The BOOT command is an alternative method for starting the spooler. Instead of using CSPOOL to invoke the CDSPL and ESPL spooler subprocesses automatically, use BOOT to start each process individually. The advantage to this is less conventional memory may be used. Ordering the commands as shown above is important.

START

Syntax: `CSPOOL START`

The START command activates the print spooler. This command should be run when the system is first booted or run after a CSPOOL QUIT command.

When the spooler starts, any spool files left over from earlier sessions begin printing. Ensure that the printer is ready to print prior to starting the spooler. See the Installing CSPOOL section later in this chapter for procedures that must be completed prior to running CSPOOL START.

QUIT

Syntax: `CSPOOL QUIT`

The QUIT command deactivates the spooler. The spooler stops printing any active spool files and all active and queued spool files are saved. With

the next CSPOOL START command, the spooler starts printing any saved spool files.

Do not QUIT CSPOOL while an application is printing and the print job has not yet been submitted to the spooler. Stop the application from printing or wait until the application is finished before using the QUIT command to stop the spooler.

CAUTION: *Always use CSPOOL QUIT to shut down the spooler prior to turning off or rebooting the CCI Concurrent DOS system to ensure that the spooler closes any open spool files.*

STATUS

Syntax: CSPOOL STATUS [C]

The STATUS command displays the status of current and pending print jobs. The jobs are displayed in the order they will be printed. If the spooler is running and there are print jobs being processed, a display like the following is shown:

```
Spooling prn#: 0,1s,2o

Job#  Process  Printer# Priority Characters State      Username User#
----  -------  -------- -------- ---------- ---------- -------- -----
   0  INCOME        0        5      42196  active     CAROL    001
   1  WP            0        5       2050  queued     JON      005
```

The fields displayed have the following meaning:

Spooled prn#	A list of spooled printers. A suspended printer is followed by an "s". An owned printer is followed by an "o".
Job#	The print job number. Use this number with the STOP and PRIORITY commands.
Process	The name of the program that created the print job.
Printer	The printer number where the job outputs.
Priority	The job's priority.
Characters	The number of characters left to print.
State	The current status of the print job. Jobs waiting for a printer are "queued". Printing jobs are either "active", "suspended", printing to an "owned" printer, or "not ready?". The "not ready" state means the spooler is trying

to print to the printer; but the printer is not ready. The printer may be turned off, off-line, or out of paper.

Username The user who printed the job.

User # The console number the job was printed from.

For a continuous status display, use the C option, as in:

```
CSPOOL STATUS C
```

The status display updates once every second until any key on the keyboard is pressed to interrupt it.

STOP

Syntax: `CSPOOL STOP [Job#]`

The STOP command deletes an active or queued print job. If a specific job number is not specified, the list of active and queued jobs is displayed with the following prompt:

```
Enter Job # to stop :
```

If the job number is already known, specify that job number in the CSPOOL STOP command. For example, to delete print job number 3 enter:

```
CSPOOL STOP 3
```

PRIORITY

Syntax: `CSPOOL PRIORITY Job# Priority#`

The PRIORITY command changes the priority level of a print job, controlling the order that the queued jobs are printed. Priorities range from 1 to 10, with 1 being the highest priority and 10 being the lowest priority. The default priority is usually 5. To change the priority of print job number 10 to priority level 5 use the command:

```
CSPOOL PRIORITY 10 5
```

Jobs are output to a specific printer in order of priority. If a print job is changed to a higher priority, that job is printed before any lower priority jobs that are in the queue. Conversely, a print job is changed to a lower priority, it is printed after jobs with a higher priority.

For example, a long accounting report was queued to a printer and then a short letter was then queued to the same printer. If it is desired to print the short letter first, use the CSPOOL PRIORITY command to make the priority of the letter's print job higher so that it prints first.

Use the CSPOOL STATUS command to display the list of currently waiting print jobs. CSPOOL STATUS displays the job number needed to use with the PRIORITY command. Look through the list of jobs and find the job numbers for the accounting report and the short letter. Use the name of

the process that created the job or the printer number the job is waiting for as clues to distinguish those print jobs from other print jobs.

Once the two print jobs have been found in the status display, use the PRIORITY command to give the short letter's print job a higher priority than the accounting report's print job. Remember, jobs with smaller priority numbers have a higher priority.

See the Spooler Configuration File section later in this chapter for information on changing the default job priority (DEFAULT_PRIORITY) and for assigning specific job priorities (JOB_PRIORITY) to specific processes.

NOSPOOL

Syntax: `CSPOOL NOSPOOL Printer#`

The NOSPOOL command tells the spooler not to spool print output for a specific printer. NOSPOOL is useful to temporarily dedicate a printer to a specific application without using the QUIT command because it stops spooling for all printers. A NOSPOOL printer is ignored by the spooler. Any applications printing to a NOSPOOL printer prints directly to the printer. For example, to stop spooling to printer number 1 (LPT2) use the command:

```
CSPOOL NOSPOOL 1
```

When a printer's spooling status is changed using NOSPOOL from the command line, that status is not retained once the spooler is stopped with CSPOOL QUIT. The next time the spooler is started with CSPOOL START, the printer will be spoolable unless it is in the NOSPOOL list in the CONFIG.CPL spooler configuration file.

The CSPOOL STATUS command displays the list of currently spoolable printers.

SPOOL

Syntax: `CSPOOL SPOOL Printer#`

The SPOOL option tells the spooler to enable spooling for a specific printer number. SPOOL is used to restore spooling for a printer that has been NOSPOOLed or to enable spooling for a printer in the NOSPOOL list in the CONFIG.CPL spooler configuration file. For example, to begin spooling to printer number 2 (LPT3) use the command:

```
CSPOOL SPOOL 2
```

When a printer's spooling status is changed using CSPOOL SPOOL from the command line, that status is not retained when the spooler is stopped with CSPOOL QUIT. The next time the spooler is started with CSPOOL START, if the printer is in the NOSPOOL list in the CONFIG.CPL spooler configuration file, it reverts to being non-spoolable.

SUSPEND

Syntax: `CSPOOL SUSPEND Printer#`

The SUSPEND option suspends spooling to a printer. This means the spooler still accumulates print output, but will not release print jobs to the printer. This is useful to temporarily stop printing. This option is reversed by the CONTINUE and RESTART options. For example, to suspend printing to printer number 3 (COM1), use the command:

```
CSPOOL SUSPEND 3
```

If a printer has been SUSPENDed and the spooler stopped using CSPOOL QUIT, all print jobs waiting for the printer will print when the spooler is started again.

CONTINUE

Syntax: `CSPOOL CONTINUE Printer#`

The CONTINUE option reverses a previous CSPOOL SUSPEND command. If a print job was printing at the time the CSPOOL SUSPEND was executed, the print job continues printing from where it left off. For example, to continue printing on printer number 3, use the command:

```
CSPOOL CONTINUE 3
```

RESTART

Syntax: `CSPOOL RESTART Printer#`

The RESTART option reverses a previous CSPOOL SUSPEND command. If a print job was printing at the time the CSPOOL SUSPEND was executed, the print job restarts printing from the beginning of the print job. For example, to restart printer number 4 (COM2) use the command:

```
CSPOOL RESTART 3
```

OWN

Syntax: `CSPOOL OWN Printer#`

The OWN option reserves a spooled printer for the use of a single console. The console running the OWN command exclusively controls the printer until a CSPOOL RELEASE command is executed. Any other application that attempts to print to an OWNed printer creates a print job that will not be released to the printer.

The OWN command is useful when running an application that should not be interrupted or when special forms (envelopes, for example) are loaded in the printer. Use the RELEASE command to allow others to use the printer when done.

For example, to own printer number 2, use the command:

```
CSPOOL OWN 2
```

RELEASE

Syntax: `CSPOOL RELEASE Printer#`

The RELEASE option reverses a previous CSPOOL OWN command. The printer is made available to other consoles. The RELEASE command must be run on the same console that executed the OWN command. Any print jobs blocked by the OWN command begin printing immediately. For example, to release printer number 2 use the command:

```
CSPOOL RELEASE 2
```

Configuring Applications for the Spooler

A multiuser, multitasking operating system like CCI Concurrent DOS can run many programs at once. For best performance, it is important that the operating system control all system resources, including printers. Many applications use operating system services for print output that allow the operating system to direct and manage all printing. These applications typically use the PRN or LST devices for printer output.

Always configure applications to print to PRN or LST when using the spooler. The built-in PRINTER command can be used to select a default printer number for the PRN device. For example, the command:

```
PRINTER 5
```

selects printer number 5 (add-on serial port number 1) as the default printer. Use PRINTMAP /P for a list of valid printer numbers on the system.

DOS applications that use DOS INT 17 interrupt calls to output to the parallel printer devices LPT1, LPT2, or LPT3 also work with the spooler. If an application needs to print to LPT1, LPT2, or LPT3 and printers are only connected to the serial ports (COM1, COM2, or an add-on serial port), install the application to use the parallel printer and then use the PRINTMAP utility to map the parallel printer that the application uses onto a serial printer port. For example, if the application prints to LPT1 and there is only a printer connected to the COM2 port (printer number 4) the command:

```
PRINTMAP LPT1=PRN4
```

maps LPT1 to PRN4. When the application sends output to parallel port LPT1, it is redirected to PRN4 instead.

CAUTION: *The PRINTER and PRINTMAP commands only apply to the printer assignments for the current session. To always use the spooler to map one printer to another, use the MAP option in the CONFIG.CPL spooler configuration file. The MAP option is discussed later in this chapter.*

Never install applications to print direct to COM devices. Applications that print to COM devices often bypass the operating system completely, out-

putting directly to the COM port hardware. This can cause unreliable behavior with the spooler.

An application that does not use operating system services for printing and insists on outputting directly to a particular printer number cannot be used with the spooler. In this case, it may be necessary to dedicate that printer to the application in question. See the NOSPOOL and NOT_PRINTER lists in the CONFIG.CPL spooler configuration file section for ways to direct the spooler not to intercept print output for a particular printer number.

If there are problems printing jobs from applications to the spooler, ask the application vendor for help in configuring the applications to use operating system services.

CSPOOL Installation

The CCI Concurrent DOS installation process copies the print spooler programs to the \CDOS directory and a creates a subdirectory called \CDOS\SPOOL. The \CDOS\SPOOL directory is the default directory where the CONFIG.CPL spooler configuration file and the spool files are stored. In addition, a CSPOOL environment variable needs to be in the environment of every session where the CSPOOL program is run. This environment variable informs the CSPOOL program where the CONFIG.CPL and spool files are located. If the CSPOOL environment variable is not present, CSPOOL does not run. Ensure the following line:

```
SET CSPOOL=%BOOTDRV%\CDOS\SPOOL
```

is in the CCI Concurrent STARTxxy.BAT, LOGIN.BAT, and/or NAMEy.BAT startup batch files. An alternative is to always run CSPOOL through a batch file, for example SPOOLER.BAT. Place a SPOOLER.BAT file in the \CDOS directory (or any other directory in your path) containing the lines:

```
SET CSPOOL=%BOOTDRV%\CDOS\SPOOL
CSPOOL %1 %2 %3 %4
SET CSPOOL=
```

This batch file sets and clears the CSPOOL environment variable every time SPOOLER.BAT is run to execute the CSPOOL program.

CAUTION: *The CSPOOL environment variable must be reset after using the CCI Concurrent DOS &SHLRESET and &ENVRESET commands. Use the* `SET CSPOOL=` *command.*

Automatically Starting the Spooler

To automatically start the spooler when the system boots, put the CSPOOL START command in one of the CCI Concurrent DOS startup batch files. The recommended procedure is to use the CCI Security System to enable the DO-UPON-BOOT option in the CONTROL file and add the following lines to the CCIBOOT.BAT file:

```
SET CSPOOL=%BOOTDRV%\CDOS\SPOOL
CDSPL BOOT
ESPL BOOT
CSPOOL BOOT
```

If using individual STARTxxy.BAT files, include the above commands in the START004.BAT file so CSPOOL START runs only on the fourth session of the host console. If using a single START000.BAT file, test for a specific console number so that the spooler only starts from a single console, as in:

```
SET CSPOOL=%BOOTDRV%\CDOS\SPOOL
IF "%1" NE "004" GOTO NOSPOOL
CDSPL BOOT
ESPL BOOT
CSPOOL BOOT
:NOSPOOL
```

When the spooler starts, any spool files left over from earlier sessions immediately start printing. Ensure that the printer paper is aligned and the printer is turned on and ready prior to starting the spooler.

Changing the Spool File directory

To change the default spool file directory, move the contents of the old spool file directory to the new directory and modify the `SET CSPOOL=` commands in the appropriate batch files to specify the new directory.

For example, when there is not enough space for spool files on drive C: so the spooler is to create spool files on drive D: in a directory called \SPOOLDIR, use the following commands:

```
C>CSPOOL QUIT ; if spooler is running
C>MD D:\SPOOLDIR
C>COPY C:\CDOS\SPOOL\*.* D:\SPOOLDIR /V/B
C>ERASE C:\CDOS\SPOOL\*.*
C>RD C:\CDOS\SPOOL
C>SET CSPOOL=D:\SPOOLDIR
C>CSPOOL START
```

Finally, update all the startup batch files that set the CSPOOL environment variable so that they use the new spool file directory. In our example, the startup batch files now contain the line:

```
SET CSPOOL=D:\SPOOLDIR
```

CAUTION: *Do* **not** *delete or attempt to access any files in the spool file directory while the spooler is running. Doing so can create unreliable behavior. The spooler assumes it has exclusive use of the spool file directory.*

The Spooler Configuration File

To configure the spooler for a particular environment, edit variables contained in the CONFIG.CPL spooler configuration file located in the spool file directory, usually C:\CDOS\SPOOL. When editing CONFIG.CPL, be sure to use a text editor or word processor that allows the use of an ASCII or non-document mode. The CONFIG.CPL variables are:

CLEAR_END_OF_LINE	Escape code to clear to end of line.
CLEAR_SCREEN	Escape code to clear the screen.
DEFAULT_PRIORITY	The default print job priority.
FORMFEED	The ASCII form feed character.
HOME_CURSOR	Escape codes to home the cursor.
JOB_PRIORITY	List of programs with special priorities.
MAP	List of printers mapped to other printers.
NOFEED_JOBS	List of programs that do not get formfeed.
NOSPOOL	List of non-spoolable printer ports.
NOT_PRINTER	List of aux ports.
TIME_OUT	Print job time out value.
HELP	List of help messages.
MESSAGES	List of error and status messages.

The following section describes the CONFIG.CPL configuration variables.

CLEAR_END_OF_LINE

Syntax: `CLEAR_END_OF_LINE [n n ...] 0`

The CLEAR_END_OF_LINE variable defines the characters (in decimal) sent to the console to clear the screen from the cursor location to the end of the line. The CSPOOL STATUS C command uses this variable to help clear the screen. This line is usually:

```
CLEAR_END_OF_LINE 27 75 0
```

which represents ESCAPE-K, the sequence used by CCI Concurrent DOS to clear to the end of the line at all stations. If this variable is not present in the CONFIG.CPL file, the CSPOOL STATUS C command clears a line by dis-

playing multiple spaces. This variable can contain up to 40 decimal numbers and must end with a zero (0).

CLEAR_SCREEN

Syntax: `CLEAR_SCREEN [n n ...] 0`

CLEAR_SCREEN defines the characters (in decimal) sent to the console to clear the screen and home the cursor. The CSPOOL STATUS command uses the CLEAR_SCREEN code to clear the screen before displaying the list of active and queued print jobs. This line is usually:

```
CLEAR_SCREEN 27 69 0
```

which represents ESCAPE-E, the sequence used by CCI Concurrent DOS to clear the screen at all stations. CLEAR_SCREEN can contain up to 40 decimal numbers and must end with a zero (0).

FORMFEED

Syntax: `FORMFEED n`

FORMFEED is the ASCII character (in decimal) that tells the printer to advance to the top of the next page. This line is usually

```
FORMFEED 12
```

which represents CTRL-L, the ASCII formfeed character.

The spooler usually prints a formfeed after every spool file to ensure that the next spool file starts printing at the top of a new page. If the FORMFEED character is set to 0 or the program that created the spool file is in the NOFEED_JOBS list, the spooler does not send a formfeed character after a print job. See the NOFEED_JOBS configuration file option described below.

HOME_CURSOR

Syntax: `HOME_CURSOR [n n ...] 0`

HOME_CURSOR defines the characters (in decimal) sent to the console to move the cursor to the home position (top left corner) of the screen. The CSPOOL STATUS C command uses the HOME_CURSOR code before updating the display of active and queued print jobs. On CCI Concurrent DOS systems, this line is usually:

```
HOME_CURSOR 27 72 0
```

which represents ESCAPE-H, the sequence used by CCI Concurrent DOS to home the cursor at all stations. HOME_CURSOR can contain up to 40 decimal numbers and must be ended with a zero (0).

DEFAULT_PRIORITY

Syntax: `DEFAULT_PRIORITY n`

DEFAULT_PRIORITY is a number between 1 and 10 that determines the default priority assigned to print jobs; 1 is high priority, 10 is low priority. This line is usually:

```
DEFAULT_PRIORITY 5
```

which is in the middle of the range of 1 to 10, allowing the setting of jobs to a higher or lower priority. A print job is assigned the default priority if it comes from a program that is not assigned a special priority in the JOB_PRIORITY list. See the JOB_PRIORITY option described below to assign specific programs special priorities.

JOB_PRIORITY

Syntax: `JOB_PRIORITY [program priority ...] END_LIST`

The JOB_PRIORITY list contains the names of applications and the print job priority to assign to print jobs when that application prints. Since spool files are output to the printer in order of priority, this controls the order that spool files are printed. The program name can be up to eight characters long and is usually the filename of the application program without the .COM, .EXE, or .CMD extension. The priority ranges from 1 to 10 (1 is high priority, 10 is low priority). For example, the line:

```
JOB_PRIORITY INCOME 1 WS4 2 SUPPORT 3 END_LIST
```

assigns the priority 1 to the INCOME application, priority 2 to the WS4 application, and priority 3 to the SUPPORT application. All other programs are assigned the DEFAULT_PRIORITY, which is usually 5. When the spooler finishes printing to a physical printer, it checks the list of print jobs waiting for that printer. The higher priority jobs are printed first. In our example, if an INCOME print job is waiting, it is printed first. If a WS4 print job is waiting, it is printed next, and so on, until the lowest priority print jobs at priority level 10 are printed.

The JOB_PRIORITY list can contain up to 40 programs and must end with the word END_LIST.

MAP

Syntax: `MAP [prn# TO prn#] END_LIST`

The MAP list is a list of printer numbers that should be mapped to other printer numbers. This is used to redirect print output meant for one printer to another printer. For example, if the line:

```
MAP 4 TO 1 END_LIST
```

is in the CONFIG.CPL file, when the spooler receives any print output for printer number 4 (COM2), it sends the print output to printer number 1 (LPT2) instead.

The MAP list can contain up to 10 mapped printers and must end with the word END_LIST.

NOFEED_JOBS

Syntax: `NOFEED_JOBS [program program ...] END_LIST`

The NOFEED_JOBS list contains the names of applications that do not require CSPOOL to send a formfeed after a print job. A program should be listed in the NOFEED_JOBS list if it prints a formfeed when finished printing. If the program prints its own formfeed and CSPOOL also prints a formfeed, a extra blank sheet of paper os printed after the print job. If the program is included in the NOFEED_JOBS list, CSPOOL will not send a formfeed after the print job.

```
NOFEED_JOBS WS WP INCOME END_LIST
```

If CSPOOL is to never print a formfeed after a print job, set the FORMFEED variable described above to 0 instead of listing all the applications in the NOFEED_JOBS list.

The NOFEED_JOBS list can contain up to 40 programs and must end with the word END_LIST.

NOSPOOL

Syntax: `NOSPOOL [n n ...] END_LIST`

The NOSPOOL printer list contains printer numbers to be exclude from the spooler. The spooler will not intercept any printer output to a printer listed in the NOSPOOL list. A NOSPOOL printer can be accessed as if the spooler is not running. For example, the line:

```
NOSPOOL 1 2 END_LIST
```

says not to spool output for printer numbers 1 and 2. A printer in the NOSPOOL list can be made spoolable while the spooler is running by using the CSPOOL SPOOL option from the command line.

The NOSPOOL list must end with the word END_LIST.

NOT_PRINTER

Syntax: `NOT_PRINTER [n n ...] END_LIST`

The NOT_PRINTER list contains the printer/auxiliary port numbers that are not used as printer ports. This parameter specifies the serial ports to use as AUX ports for auxiliary devices such as modems. For example, the line:

```
NOT_PRINTER 3 4 END_LIST
```

defines printer number 3 (COM1) and printer number 4 (COM2) as NOT_PRINTER ports since COM1 and COM2 are commonly used as AUX ports for modems or as mice ports and are not used as printer ports.

The NOT_PRINTER list must end with the word END_LIST.

TIME_OUT

Syntax: `TIME_OUT n`

TIME_OUT instructs the number of seconds the spooler waits before automatically detaching a program from the printer. This is for DOS applications that do not interface with CCI Concurrent DOS system services to detach from a printer. If the spooler detects a program has not been trying to print characters for TIME_OUT seconds, it assumes the program is finished printing, detaches the program, and submits the print job to the spooler. TIME_OUT can range from 1 to 9,999 seconds. This line is usually:

```
TIME_OUT 10
```

which specifies a TIME_OUT of 10 seconds. If the programs pause for long periods of time between sending print characters (as in some graphics applications) increase the TIME_OUT value. If the programs do not have a long delay between printed characters, decrease the TIME_OUT value.

If the TIME_OUT value is too high there may be an unacceptable delay between the time a program finishes printing and the time the spooler outputs the spool file. If the TIME_OUT value is too low, the spooler may assume a program is finished printing before it really is, and submit several individual print jobs for a single printout.

A TIME_OUT value of 5 to 15 seconds is recommended.

HELP, MESSAGE, Comments, and END

Syntax:
```
HELP...END_LIST
MESSAGE...END_LIST
;This is a comment
END of CONFIG.CPL
```

The HELP and MESSAGE lists should not be modified without the advice of the Concurrent Controls technical services team. These lists contain status and information messages displayed by the CSPOOL program.

Comments can be placed in the CONFIG.CPL file after a semicolon (;). The CSPOOL program ignores all characters in the CONFIG.CPL file between a semicolon and the end of the line.

The last line in the CONFIG.CPL file must begin with the word END. Anything after a line beginning with END is ignored.

CSPOOL Error Messages

The CSPOOL program displays the following status and error messages.

▼ `Can't allocate flags for ...`

This message occurs during CSPOOL START if the spooler cannot find a free system flag to use. The spooler uses flags to reduce overhead while the spooler is dormant (that is, the spooler is running but nothing is being printed). For some reason, CSPOOL tried to allocate a flag and was denied. This is not a fatal error. If the spooler cannot use flags it uses delays instead. Using delays slightly decreases overall system performance.

▼ `Can't attach to printer ...`

When `CSPOOL SPOOL` was issued to re-enable spooling for a particular printer, CSPOOL tried to gain control of the printer by attaching to it and failed. An invalid printer number might have been specified or the printer might be in use by another process. Use PRINTMAP /P to display a list of valid printer numbers or run SYSTAT [A] to see if the printer is currently being used by someone else.

▼ `Can't change printer to ...`

For some reason, a CSPOOL SPOOL/NOSPOOL or CSPOOL SUSPEND/RESTART/CONTINUE command has failed. The next line should describe the error in more detail.

▼ `Can't find the CSPOOL variable ...`

The CSPOOL program looked in the current environment and did not find the CSPOOL environment variable. CSPOOL needs the environment variable to find the CONFIG.CPL file. Use SET to see if CSPOOL is defined in the current environment. If not, check the SET CSPOOL= statements in the startup batch files.

▼ `Can't rename ...`

When CSPOOL starts, it looks in the spool file directory for print jobs that were active or were queued when a CSPOOL QUIT (or system reset) occurred. CSPOOL renames these files so it can re-spool them. For some reason, this renaming process has failed.

▼ `CONFIG.CPL: invalid value ...`

▼ `CONFIG.CPL: invalid process name ...`

▼ `CONFIG.CPL: expected \TO\ keyword ...`

▼ `CONFIG.CPL: the option is missing ...`

▼ `CONFIG.CPL: unexpected end of file ...`

▼ `CONFIG.CPL: too many entries in ...`

▼ `CONFIG.CPL: unknown keyword ...`

These messages point out problems with the CONFIG.CPL spooler configuration file. Edit the CONFIG.CPL file to fix them.

▼ `CONFIG.CPL: ran out of memory ...`

CSPOOL ran out of memory while reading the HELP or MESSAGE list. Maybe MEMSIZE is too low or CSPOOL is PIFEDed for too small a MINMEMORY value. Try freeing up memory before running CSPOOL START again.

▼ `Installing Spooler...failed! ...`

CSPOOL START failed.

▼ `Invalid printer number ...`

A CSPOOL SPOOL, NOSPOOL, SUSPEND, RESTART, or CONTINUE command was issued with an invalid printer number. This error message displays a range of numbers that are valid printer numbers. This does not mean that every number in that range is a printer number. Use PRINTMAP /P to display a list of valid printer numbers for the system. Use SETUP and/or CONFIG to verify that a port is usable as a printer port.

▼ `Job Priority changed.`

CSPOOL PRIORITY has successfully changed a print job's priority.

▼ `Not enough command line arguments.`

An incomplete CSPOOL command line has been entered. For example, `CSPOOL NOSPOOL` was entered but a printer number was not supplied or `CSPOOL PRIORITY` was entered without a job or priority value.

▼ `Printer already set to ...`

A CSPOOL SPOOL, NOSPOOL, SUSPEND, RESTART, or CONTINUE command was issued and the printer was already in the desired state. For example, `CSPOOL SPOOL` was entered and the specified printer is already a spoolable printer.

▼ `Printer changed to ...`

This message occurs after a successful CSPOOL SPOOL/NOSPOOL or CSPOOL SUSPEND/RESTART/CONTINUE command.

- `Printer is being used.`

  The CSPOOL NOSPOOL command was entered for a printer that is currently printing a print job. Wait for the printer to stop printing before using CSPOOL NOSPOOL.

- `Printer is a NOSPOOL printer.`

  `CSPOOL SUSPEND` was entered on a printer that is not being spooled.

- `Printer is a NOT_PRINTER port in CONFIG.CPL`

  There was an attempt to change the SPOOL or SUSPEND status of a printer number in the NOT_PRINTER list in the CONFIG.CPL spooler configuration file. If a printer is in the NOT_PRINTER list, the spooler cannot use it. If the printer really is a printer, it should not be in the NOT_PRINTER list. Use PRINTMAP /P for a list of valid printer numbers for the system.

- `Printer is not suspended.`

  `CSPOOL RESTART/CONTINUE` entered for a printer That had not been previously SUSPENDed.

- `Priority value must be between 1 and 10.`

  `CSPOOL PRIORITY` used with an invalid priority value.

- `Specified job stopped and removed from spool list.`

  This message occurs when CSPOOL STOP successfully stops a job.

- `Spooler is Running`

  This message occurs if `CSPOOL START` is issued when the spooler is already running.

- `Spooler not Running`

  This message occurs if you run any CSPOOL command (except for CSPOOL START) and the spooler is not running. CSPOOL must be STARTed first.

- `Terminating Spooler...failed! ...`

  This message displays if CSPOOL QUIT fails.

- `Unable to change that job's priority.`

  This message occurs when CSPOOL PRIORITY is used with a nonexistent job number or with the job number of a print job that is already printing. Since the job is already printing, it does not make sense to

change it to a higher priority and it is too late to change it to a lower priority.

▼ `Unable to change to CSPOOL ...`

CSPOOL could not set the drive or directory specified in the CSPOOL environment variable. Check the value of the SET CSPOOL= statement in the startup batch files. This could also happen if a drive is not specified in the CSPOOL environment variable. If a drive is not specified, CSPOOL looks for the spool file directory on the current drive.

▼ `Unable to open the CONFIG.CPL file`

An error has occurred while trying to open the CONFIG.CPL file. Ensure that the SET CSPOOL= environment variable is correct and the CONFIG.CPL file exists in that directory.

▼ `Unable to stop that job number.`

This message occurs when trying to CSPOOL STOP a job number that does not exist. Run CSPOOL STATUS or CSPOOL STOP (without a job number) for a list of current job numbers.

▼ `Unknown command line argument`

CSPOOL was given an unknown command line argument. Use CSPOOL HELP to display a list of valid commands. This usually happens because of a misspelled CSPOOL command.

▼ `There must be AT LEAST 1 MB of free disk space ...`

This message occurs when there is less than one megabyte of disk space available on the drive where the spool file directory is located. Either free up disk space on the current CSPOOL drive or move the spooler directory to another drive.

This page intentionally left blank.

Index

Index

!

&CLEARCCI 3-4
&SETCCI 3-1, 3-4

A

AND 3-11
Attributes 1-6, 4-15

B

Batch commands 3-11, 3-13, 3-15, 3-17, 3-19
Batch file variables 3-8
Batch files 3-1
  Creating 3-3

C

CALL 3-12
CASE 3-12
CCI Concurrent DOS 1-1
CCI NET 1-1
CCI TERM 5-28–5-29
COM1 5-28
COM2 5-28
COM3 5-28
COM4 5-28
Command line 1-14–1-15
Commands 1-10–1-11, 1-13
Configuration
  CSPOOL 8-8
  CCI Concurrent DOS 2-5, 2-14–2-15
  Host computer and system console 2-6
  Modem 2-12
  Mouse 2-13
  Multiuser serial ports and display adapters 2-9
  Printer 2-13
  System memory 2-8
CONTROL 3-1, 4-6
  File options 4-10
CSPOOL
  Commands 8-2–8-3, 8-5, 8-7
  CONFIG.CPL 8-11, 8-13
  Configuration 8-8
  Configuration file 8-11, 8-13
  Continue 8-7
  Error messages 8-15, 8-17, 8-19
  Excluding printers 8-14
  Installation 8-9
  Nospool 8-6
  Own 8-7
  Priority 8-5
  Quitting 8-3
  Release 8-8
  Restart 8-7
  Spool 8-6
  Starting 8-3
  Status 8-4
  Stopping jobs 8-5
  Suspend 8-7
CTRL-\ 1-16
CTRL-_ 1-16
CTRL-A 1-15
CTRL-B 1-15
CTRL-Break 1-14
CTRL-C 1-14
CTRL-D 1-15
CTRL-E 1-15
CTRL-G 1-15–1-16
CTRL-H 1-15
CTRL-J 1-15
CTRL-K 1-15
CTRL-L 1-14
CTRL-M 1-15
CTRL-n 1-14
CTRL-O 1-14
CTRL-P 1-14
CTRL-Q 1-14–1-15
CTRL-R 1-15
CTRL-S 1-14–1-15
CTRL-T 1-15
CTRL-U 1-15
CTRL-V 1-15

CTRL-W 1-15
CTRL-X 1-16
CTRL-Y 1-16

D

DATE 3-9
Dealer starter kit 2-3
Defragmentting a disk 2-18
DIREXIST 3-12
DOWN-ARROW key 1-16

E

ECHO 3-13
ECHOERR 3-13
END key 1-16
ENTER key 1-15
Error messages 7-7, 8-15, 8-17, 8-19
EXIT 3-13

F

FDISK utility 2-3
File
  Open 5-46
  Attributes 1-6, 4-15
  Naming guidelines 1-5
  Permissions 4-12
  Security 1-7, 4-11, 4-13, 4-15
  Groups 4-12
  Transfer 5-28
FLAGWAIT 3-3
Floating drives 1-8
FOR 3-14
FORCE 3-3
Formatting a disk 2-3
Function keys 1-16

G

GOSUB 3-15
GOTO 3-14
Groups 4-12

H

Help 1-17
HISTORY.DAT 4-10–4-11
HOME key 1-15
Host computers 1-2
Host consoles 1-2

I

IF 3-15
INPUT 3-17
INPUTC 3-17
INPUTX 3-17
Installation
  CSPOOL 8-9
  CCI Concurrent DOS 2-1, 2-4

L

LOADER.COM 2-4
LOADSYS.COM 2-4
LOGIN.BAT 3-2
LOOKOUT 5-19–5-23
  REMOTE and TAKEOVER 4-1, 5-20–5-23

M

MDA
  see Multiuser display adapter
Modem configuration 5-4
Multitasking 1-1–1-3
Multitech V32 9600 baud Modem 5-5
Multiuser 1-1–1-2, 1-4
Multiuser display adapter 1-1
Multiuser serial boards 1-1

O

ON 3-17
Optimizing CCI Concurrent DOS 2-17
OR 3-18

P

Partitioning a hard disk 2-3
Password 4-5
Passwords 1-7, 4-2–4-3, 4-5
Pathing 1-9
PAUSE 3-18
PAUSEERR 3-18
PCTERM stations 1-2
PDELAY 3-3, 5-35, 5-45

Permissions 4-12, 4-16
PFK
see Programmable function key
PGDN key 1-16
PGUP key 1-15
Piping 1-17
Print spooler 8-1
See also CSPOOL
PrintScrn key 1-14
Programmable function keys
1-16

R

Redirection 1-16
REM 3-18
REMOTE 5-19–5-20
access 5-20
See also LOOKOUT
Remote access 5-3
Remote workstations 5-1
Removable media 1-9
RETURN 3-19

S

SAMPLE.BAT 3-3, 3-8
SAMPLE.EXE 3-3
SCHEDULER 5-28
Security 1-7, 3-1, 5-3, 5-23
CONTROL 4-6
Security System 5-23
AUTO-LOGIN 4-8
CLEAR-BUFFERS 4-9
CONTROL 4-6
DO-UPON-BOOT 4-8
DOUBLE-BLIND-LOGIN 4-9
Enabling 4-1
FILE SECURITY 4-8
HISTORY 4-2, 4-9
INCLUDE 4-10
MULTI 4-8
NO-CLEAR-BUFFERS 4-9
PASS 4-2–4-3, 4-5
PASSWORD 4-8
SECURITY 4-1
SINGLE 4-8
START-FIRST 4-10
Serial ports 5-4
Sessions 1-2
SHIFT 3-19
Spooler
See also CSPOOL
START.BAT 8-10
STOP 8-5
SWITCH 3-20
SYSADMIN.DAT 4-10–4-11

T

TAB key 1-14
TIME 3-9
TRANSIT 5-28–5-29
Script errors 5-44–5-45, 5-47
Script language 5-32–5-33,
5-35, 5-37, 5-39
TRANSIT script 5-28
Troubleshooting
Error messages 7-7, 8-15, 8-17,
8-19

U

UTMP.DAT file 4-2

W

Wildcards 1-7
Workstations 1-2, 5-1

X

XATTRIB 4-15–4-16
XMODEM 5-28

Y

YMODEM 5-28

Concurrent Controls, Inc.
349 Allerton Ave.
S. San Francisco, CA 94080
650-873-6240
Sales: 800-487-2243
Fax: 650-873-6091
BBS: 650-873-6256
Internet: info@conctrls.com